Umeå Universitet
Department of Computer Science
Umeå, Sweden

180505

"Visions and delimitations"

Security and convergence in Voice over IP

Master Thesis Project corresponding to 20 credits by
Jens Tinglev <ens00jtv@cs.umu.se>
Written and developed at Teleca Software Solutions AB, Umeå

Internal supervisor: Pedher Johansson <pedher@cs.umu.se>
External supervisor: Fredrik Granström <fredrik.granstrom@obigo.com>
Grader: Per Lindström <perl@cs.umu.se>

Abstract

This master thesis project aims at giving the reader an insight into the area of security and convergence in voice over IP systems. The main goal of the thesis is to provide the reader with an overview of the security risks and the measures of defense involved in voice over IP. It also provides a short market analysis provided for TSS Umeå, the corporation for whom this thesis was written. Besides the theoretical analyses performed, a pilot application was developed to test the possibility of converging the GSM network with the PSTN through a Bluetooth© link. The thesis further discusses various types of possible convergence and also the future for the technology as a whole.

Acknowledgements

First of all I would like to thank all of the people at TSS Umeå for giving me the opportunity to work with them. Special thanks go out to Rolf Sandberg for making me a part of the TSS family (if only for a couple of months), going along with my ideas and providing feedback throughout the writing process. Fredrik Granström, my supervisor at TSS Umeå, also deserves big thanks for all the valuable help he has provided during these 18 or so weeks. I would also like to give a shout out to Jonas Andersson (also at TSS Umeå) for giving me some well needed time off from my writing for a little scripting. Of course I've saved a thank you for my internal supervisor, Pedher Johansson, whose optimistic attitude and motivational meetings kept me going strong. Finally I would like to thank my girlfriend Therese who probably made more of a difference than I've recognized, keeping me motivated all this time and listening to my, sometimes seemingly endless, brainstorming monologues.

Table of contents

List of Tables

List of Figures

1 Introduction

Ever heard of voice over IP? Even though it is a technology that has been around for quite some time, do not feel bad if you answered "no" to the previous question. Late user adoption and market acceptance has made voice over IP technology a shy debutant to the world of communication. As of today, not much scientific work exists in the area of voice over IP, and even less on the security concerning it. Few attempts of merging different networks of communication with each other have been successful. This seems remarkable considering the revenues possible with this technology.

Voice over IP is the result of an attempt of convergence - the idea of converting two networks of communication into one single, high performing medium of data transportation. As with all technology, the demand for this convergence lies in the hands of the consumers. The consumers, and therefore, the market, could never have found interest for this convergence if there had not been money to make by migrating to the new technology. This give and take relation between developers of the technology and the market embracing it could never come full circle if the network and data passing through it was not secure. With such an evolutionary new technology introduced, both the stakes and demands are high. A justifiable question is of course if this is an unreachable utopia. Is the idea of a fully converged network that is making money for consumers, operators and developers while fulfilling all different parties' demands for functionality, quality of service, security and availability even viable?

Although some question marks remain, the interest in this new technology and the possible money to be earned from it, still attracts quite few actors on the market. The new kind of functionality possible with increased bandwidth and close interrelation with computers also serves as an attractant to more and more corporations wanting their fair share of success. The risk is of course that the development and deployment of VoIP solutions will escalate at an uncontrollable rate, finally being caught up by the reality of QoS and security issues.

Convergence is the vision, security is the delimiter and the market is the jury of voice over IP technology as communication tries to take another step further into the future.

1.1 Background to voice over IP security

Today's telephone communication could very generalized be categorized into three different large systems. The first is the airborne mobile net that is used by wireless mobile communication devices, such as mobile phones. The second one is the static, wire connected PSTN (Public Switched Telephone Network), often referred to as the "static net". This is what we use when we make calls on our regular phones from your home to someone else's home (or where ever there is another regular phone). These two first systems are interlinked, making it possible to make a call from one system, ending up at the endpoint of another system. With the emersion of the Internet and the discovery and exploration of its potential, the need for more and faster connection to it arose. Today this network is incredibly large and used by millions and millions of people on a day to day basis. This is our third system – the packet based IP network.

Voice over IP (VoIP) is one step closer to the convergence between the IP network and the PSTN. Moving calls away from the PSTN and on to the IP network is a way to cut costs. Not only will calls be less expensive, as seen in Section 3, to make using the IP network, but one network doing two networks job can not be seen as something other than major progress.

Sending voice over IP-networks is a packet based service, in contrast of the signal bound PSTN (actually the PSTN has capabilities of sending data as well, although on a different physical network [3]) which makes VoIP suitable for more elaborate solutions, such as incorporating multimedia and intelligent services.

VoIP sends information in a similar manner as, for example, your Web browser, cutting the information into smaller parts, putting the parts into packets, adding headers, and shipping them off across the network. The technique of sending data in packets as emails or http-requests has been around for quite some time and good mechanisms for security have already been developed [4]. VoIP is not only special in its quality of service (QoS) requirements (that includes, among other things, low latency and almost 100% availability, but it also has its own infrastructure which creates new potential targets for an attack [2]. As VoIP uses packets to send data over possibly unmonitored and insecure networks it opens up for a whole new range of security issues that the PSTN did not have.

Security in VoIP systems has a tendency to be neglected as the corporate demands of QoS issues, interoperability and cost efficiency puts pressure on the providers of the technology. Although the strides of development concerning these areas of interest could prove to be an invaluable asset for the development of VoIP, one can not overlook the importance of security in such a accessible system [2].

1.2 Purpose and outline

The purpose of this report is to investigate various threats and security measures regarding the VoIP technology and to evaluate different ways to achieve convergence between different networks. It also contains a short market analysis which purpose is to evaluate Teleca Software Solutions possible economical benefits by migrating to VoIP. Besides from this, a prototype piece of software has also been developed, exploring the possibilities of further convergence between two different communication networks.

In Chapter 2 there is a short structural analysis of the differences between the existing publicly switched telephone network and an IP based telephone network.

The economical factors of voice over IP are discussed in Chapter 3, which contains a market analysis requested by TSS Umeå. This chapter is intentionally kept brief because of the report's focus on the technical aspects of voice over IP technology and the analysis is focuses specifically on TSS Umeå and its requests.

Chapter 4 covers the current issues in voice over IP security and suggestions on how to defend systems against them. Because of the possible span of this topic, the chapter's content is width- instead of depth oriented.

Chapter 5 evaluates and discusses the VoIPBT application. The chapter also includes evaluations of different ways to achieve convergence. Both different programming languages are evaluated, as well as different technologies (such as infrared and Bluetooth©).

2 A structural analysis

Before any actual analysis of different threats and defenses can be made, it is important to understand the infrastructural differences between the PSTN and the network used by VoIP. As the two networks consist of different network entities and also have two very separate structural designs, the possibilities of attacks differs largely.

2.1 PSTN architecture

In Figure 1 the architectural layout of a traditional telephone network is shown. The most important thing to notice in the figure is the separation between the data and telephone lines.

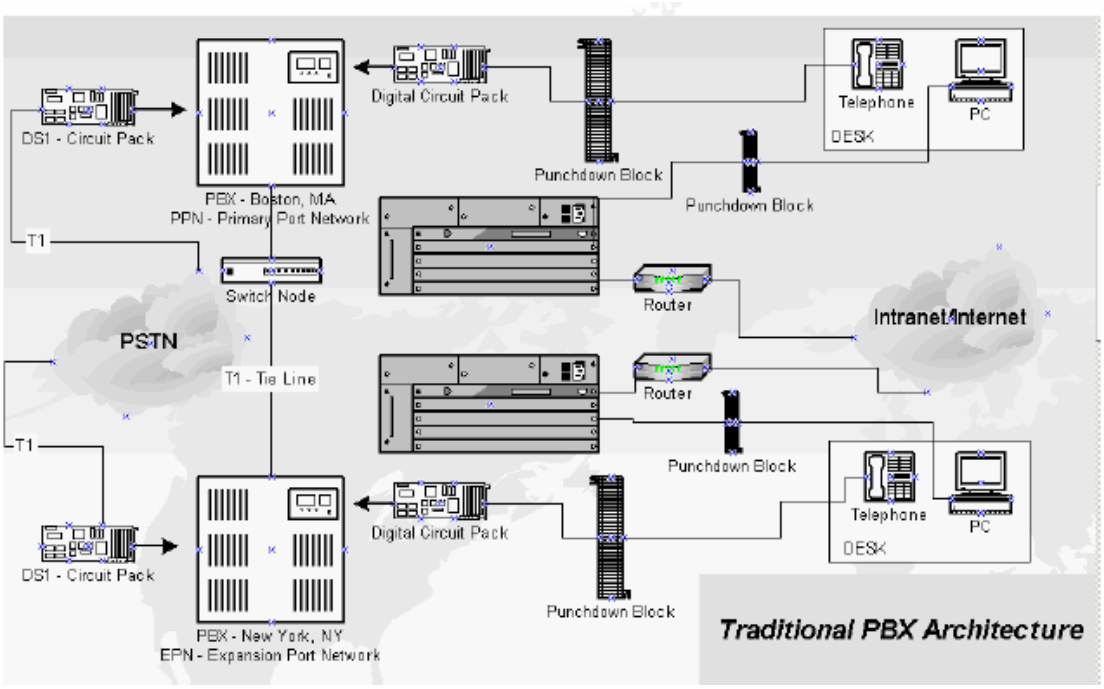

Figure 1 [2]. Traditional PBX Architecture

Each telephone has a designated connection to the PBX. At the PBX each incoming connection has a static mapping to a physical port. Static mapping of ports is part of what makes the PSTN a static network when compared to the VoIP. The two PBXs are connected to each other by a designated T1-connection. This is also a static type of design (as will be seen in the section VoIP architecture) [2].

2.2 VoIP architecture

A detailed topology of a VoIP network is displayed in Figure 2. In this figure the telephone and data networks has been merged into one single packet bound network.

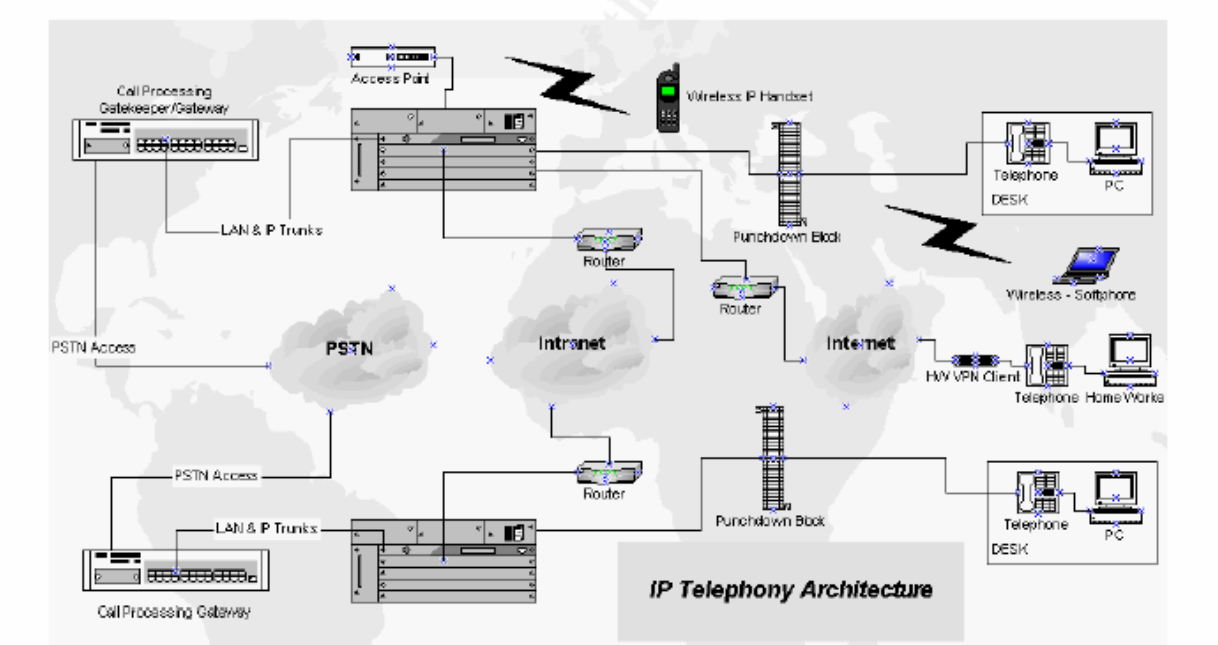

Figure 2 [2]. IP Telephony Architecture

Instead of having static connections between the telephones and the PBX:s the connection is now much more dynamic, running over an Ethernet network. Worth noticing is also that the dedicated T1-line that used to connect the two PBX:s has been replaced by IP-trunks.

2.3 SIP

SIP is a text-encoded request-reply protocol (it actually has large similarities with HTTP [9]) which makes is less complex [8] and highly extensible [7]. SIP is also somewhat of a de facto standard, getting much more attention in articles and literature then other standards. This, along with the fact that SIP is widely supported in products by numerous large corporations such as Cisco (e.g., in the Cisco IP Phone 7960 and Cisco IOS® Gateway [6]) makes it especially interesting to take a closer look at. As SIP is so widely used in VoIP systems, it is relevant to inspect the protocol a little closer to further understand the security risks involved with VoIP.

The Session Initiation Protocol[1] was designed by the Internet Engineering Task Force[2]. Cisco defines SIP as:

> *".. a peer-to-peer, multimedia signaling protocol that integrates with other Internet services, such as e-mail, Web, voice mail, instant messaging, multiparty conferencing, and multimedia collaboration. When used with an IP infrastructure, SIP helps to enable rich communications with numerous multivendor devices and media."[6]*

SIP should be seen as a component that in combination with other IETF protocols is used to provide complete services for the user. Although this is the preferred way to use SIP, the basic functionality and operations of SIP still work if separated from other protocols [8]. SIP is, as its name implies, a protocol for session initiation. SIP doesn't know anything about the sessions it handles, except for how to initiate them. A widely used protocol for describing the actual session and its parameters is the Session Description Protocol[3]. In fact, SIP includes mechanisms for negotiating a protocol for describing sessions between parties so that SDP necessarily doesn't have to be used [9]. SIP basically has four major components, namely:

1. The user agent
2. The proxy server
3. The redirect server
4. The registrar

2.3.1 The user agent

The user agent is the endpoint entity handling the initiation and termination of sessions. These entities could be an IP phone, a conferencing server and more [6]. The user agent act much as a regular phone does in the PSTN, initiating calls[4] and terminating them[5].

2.3.2 The proxy server

The proxy server acts as both a server and a client as they route traffic from user agents to requested destinations. Requests by user agents can both be handled internally within the proxy or be distributed to other servers for further processing [7].

The proxy accepts invite requests from the user agent and passes these on to the requested endpoints. The proxy server can also convert a single invite to multiple instances[6] to try to initiate contact with several user agents at once. This forking process could be seen as when all phones at someone's house ring at once, instead of just one of them. This effect is in VoIP caused by the proxy server forking an incoming invite to all that persons registered phones [9].

---

[1] SIP, IETF RFC 3261
[2] IETF
[3] SDP, RFC 2327
[4] Dialing a number
[5] Hanging up
[6] Also called "forking"

2.3.3 The redirect server

The redirect server should be seen as a proxy that doesn't forward requests. The redirect server accepts requests from proxies, but instead of forwarding the requests to the next proxy or to the final user agent it instead replies to the proxy with a special redirect response. The redirect server store databases over known users and looks these up upon request from proxies. As the proxy receives the redirect reply from the redirect server it now knows where to redirect its initial invite request [9].

2.3.4 The registrar

The registrar work in close correlation with the proxy servers to register new contact information, like for example mapping SIP addresses to IP addresses, and storing it in a database that the proxy server then can access when it needs to know where to send it's invite requests (much like a DNS lookup). The registration process is made with the register request sent by the proxy [7].

3 A market analysis

Convergence has its very own meaning in the economical aspect of VoIP. Not only is it limited to convergence between different mediums of transportation, but also by convergence of data sent. This is for example convergence of voice, video and data onto one network. Not only does moving all communication onto one channel of transportation save telephone costs, it also saves infrastructural costs [40].

In order to provide information about the call cost efficiency by using VoIP instead of regular telephony, a comparison between different providers of telephony is provided in Table 4. These providers were chosen based on different basis. TeliaSonera was chosen because it being a huge PSTN service provider in Sweden, owning the local loop and thus charging subscribers for this critical resource. The subscription form used in the comparison is their regular business telephony. Bredbandsbolaget was chosen because of its IP-telephony solution and because of it's nationally used Internet services. They provide a hardphone VoIP solution through signal conversion. A small box converts the regular telephones signals to data packets before shipping them of through the internet socket. The final operator, Skype, was chosen because of its enormous popularity (more than 100 million downloads so far [26]). Skype provides softphone VoIP by offering free software and a subscription form called SkypeOut. Through SkypeOut, subscribers can call IP-to-IP for free, but also regular telephones at reduced rates. To clarify the results of this comparison, the rates for the different providers and services used in the comparison are provided in Table 1, 2 and 3. All prices include Swedish sales taxes.

TeliaSonera Rates (23/3 2005)	Service: Företag (8-18, mon-fri)
Destination	**SEK / min**
National Call (Sweden)	0.184
Mobile (TeliaSonera)	2.000
Mobile (other)	2.360
International Call (France)	1.200
International Call (Germany)	0.950
International Call (China)	6.450
International Call (Taiwan)	3.150
International Call (Japan)	3.150

Table 1 [28]. Rates for TeliaSonera Företagsabonnemang, calls made between 8am and 6pm

Bredbandsbolaget Rates (23/3 2005)	Service: IP-Telefoni
Destination	**SEK / min**
National Call (Sweden)	0.190
Mobile (TeliaSonera)	2.250
Mobile (other)	2.350
International Call (France)	0.830
International Call (Germany)	0.830
International Call (China)	5.800
International Call (Taiwan)	4.460
International Call (Japan)	3.150

Table 2 [29]. Rates for Bredbandsbolaget IP-teleponi service

Skype Rates (23/3 2005)	Service: SkypeOut
Destination	**SEK / min**
National Call (Sweden)	0.152
Mobile (TeliaSonera)	2.216
Mobile (other)	2.216
International Call (France)	0.175
International Call (Germany)	0.175
International Call (China)	0.227
International Call (Taiwan)	0.227
International Call (Japan)	0.172

Table 3 [27]. Rates for SkypeOut

Comparison			
Destination/Provider	**SkypeOut**	**TeliaSonera Företag**	**BBB IP-Telefoni**
National Call (Sweden)	80%	97%	**100%**
Mobile (TeliaSonera)	98%	89%	**100%**
Mobile (other)	94%	**100%**	99%
International Call (France)	15%	**100%**	69%
International Call (Germany)	18%	**100%**	87%
International Call (China)	4%	**100%**	90%
International Call (Taiwan)	5%	71%	**100%**
International Call (Japan)	5%	**100%**	**100%**

Table 4. Price comparison between the different providers. Cells marked 100% are the most expensive solution for that particular destination. The other percentages show how the price of those respective providers compares to the most expensive for that destination.

Another factor that needs to be included in economical calculations is the start-up costs for migration to VoIP systems. A research performed by Nemertes revealed that startup-costs range from $512 to $1,512 per user [30]. The startup-cost includes such things as IP-PBXs, headsets and equipment upgrades but also assessment, planning, installation and troubleshooting. These numbers applied to TSS Umeå[7] would mean a startup cost from somewhere between around 240,000 to 700,000 SEK[8]. These numbers may be very incorrect if the purpose for migration is not to increase productivity through better infrastructure and customer support (many of the big provider's solutions are aimed at this). In some cases, subscribing corporations are just interested in lower call costs with the possibility to incorporate more functionality in the future. Corporations that are only interested in cutting down call costs might be good of with a single PBX and an IP-telephone for each employee. Some corporations are aiming at such easy migrations as using VoIP software to cut down telephony costs. An interesting aspect is the possible cost reduction for TSS Umeå by migrating to Skype instead of stationary phones.

3.1 TSS Umeå and Skype

The current number of employed people at TSS Umeå is 65, each with their own personal mobile phone to use at work. There are also 15 stationary telephones used for telephone meetings etc. The call costs for the mobiles are about 1200SEK/quarter per mobile and the stationary telephones call costs are around 2000SEK/quarter per phone. Total costs for telephony per quarter are 108,000SEK. Some assumptions can be made regarding the telephony at TSS Umeå[9]:

1. 30% of all telephony calls are made from office to office (national calls).
2. 50% of all telephony calls are international calls (exchangeable with IP to IP).
3. 20% of all telephony calls are international calls (exchangeable with IP to Local network)

[7] The number of people currently employed at TSS Umeå is 65
[8] Exchange rates from http://www.forex.se at 290305
[9] Assumptions provided by Rolf Sandberg at TSS Umeå via mail 080405

Based on these assumptions the call costs can easily be heavily reduced by migrating to Skype. Both category 1 and category 2 in the above list can be totally eliminated by using Skype, while the cost for the calls that fall under category 3 can be cut to a minimum. If both category 1 and category 2 are eliminated, 80% of all costs are removed making the total cost of telephony 21,600SEK/quarter. International calls made by SkypeOut costs, according to Table 4, in average around one tenth of the other providers. Mapping this cost onto the remaining telephony costs of TSS Umeå, after the first reduction, only 2030SEK/quarter remains. This is about 2% of the current telephony costs for TSS Umeå.

These kinds of cost reductions are, however, a utopia for corporations such as TSS Umeå. Mobile calls can not be replaced by headsets connected to a computer running VoIP software. Corporations are also too dependent on performance, availability and quality of service to entirely migrate to a free piece of software handling all their communication. If these obstacles could be passed, then the interest in the VoIP technology would probably skyrocket. The solution to the first problem, regarding the freedom of movement, might be solved by such programs as the one developed in this master thesis project. More about the development of Voice over IP via Bluetooth© and convergence in particular can be found in Chapter 5.

4 Security in Voice over IP networks

The federally funded research and development CERT® Coordination Center (CERT/CC) stated that 3,780 vulnerabilities were reported in 2004 [10]. This number is almost equal to 2003's 3,784 vulnerabilities. Statistics like these show that computer security is not something that will go away on its own but is rather something that needs to be taken into consideration when dealing with computers. Security does not take in to consideration whether it is a single workstation at someone's home or if it is a LAN encapsulating 200 computers at a large corporation.

VoIP terminals are made up by softphones[10] and hardphones[11]. The softphones are dependent on the computer they are running on to be able to provide the VoIP service. If the computer doesn't work, neither does the softphone. The hardphone, on the other hand, is like a regular telephone completely dependent free and works with or without a computer. Hardphones can be extended with functionality that represents that of a computer. As an example, most of today's Cisco® 7900-series of IP-telephones support software upgrades through a built in TFTP and some also include support for the XML [11]. The Snom® 220 IP-telephone even has an integrated HTTP server [12].

Security in the softphones is trivial. The softphone will be just as vulnerable as the computer it runs on and can even make the computer less secure if it, in itself, acts as a security risk by containing implementational bugs or perhaps opening ports that provide a way into the system for attackers.

Hardphones are standalone with their own firmware and circuitry. Integrated software that comes with hardphones is a bittersweet trend. Functionality increases, as does the possibilities of upgrading software, eliminating security flaws and keeping protocols up

[10] Software that emulates a telephone on a computer
[11] Hardware telephones that support the VoIP technology

to date. On the other hand, functionality and extended support for software possibly creates a much more susceptible environment for attacks.

As both hardphones and softphones are software dependent (hardphones has its firmware and softphones is a piece of software) and they both use IP networks to communicate they are both targets to a wide variety of different attacks and exploits. Although there are some similarities between the two it is crucial to note the above differences as they impact on how well attacks work on the two types of telephones.

4.1 Denial of service attacks

Denial of Service[12] is the group name for different types of attacks all flooding some network entity (such as a server, router or terminal). Large amounts of information forcing the target to make some sort of computation (for example I/O operations, work for the CPU or bandwidth consuming activities – the heavier the load the "better") are send to a target, hopefully making it choke on the amount of information and eventually crash. DoS also have a "big brother" in the Distributed Denial of Service[13] attacks. The difference between the two types is the number of sources launching the attack. DoS attacks has a single point of origin whilst DDoS attacks has multiple (often in the hundreds or even thousands) points of origin used for a coordinated attack against a common target [25].

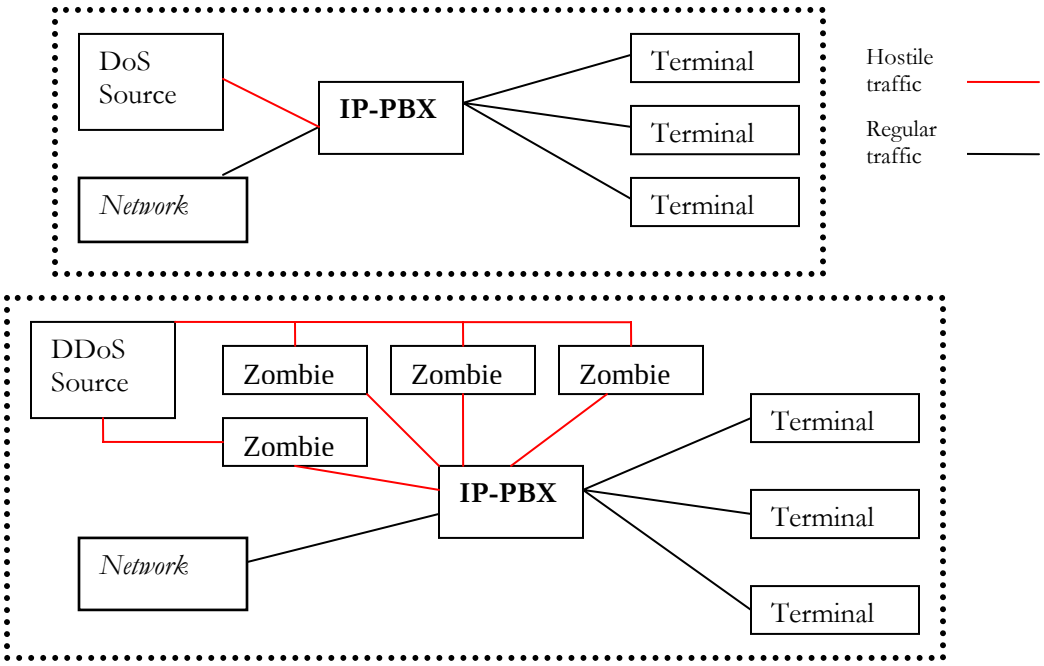

Figure 3. An overview of DoS and DDoS attacks. The hostile traffic results in the IP-PBX inability to serve the terminals with requests and to access the network properly. Enough load cause by this hostile traffic will force the IP-PBX to entirely shut down, making it impossible for the terminals to phone out or for people to phone in.

[12] DoS
[13] DDoS

As VoIP is extremely QoS sensitive (especially in the aspect of latency) it is a favorable target for DoS[14] attacks. Latency in VoIP means delay between the input given at the sender and the speech heard at the receiver. This latency varies due to distance data has to travel, overhead caused by various network entities passed on the way to the receiver, packet size of the data sent and also the operations[15] carried out by the terminals itself [29]. The attacker doesn't have to entirely crash the VoIP service; it's enough to make the target unable to communicate within a reasonable amount of time. As both softphones and hardphones in a home or office depend on the same bandwidth for functionality, an attack that targets that connection can effectively create problems for either of these two terminals.

The same approach of stopping DoS attacks currently used on the Internet must be adopted by network engineers to effectively fight of the possibilities of these kinds of attacks launched against VoIP. This includes proper configuration of routers, servers and firewalls as well as the overall design of the network [1]. Defence against DoS is much simpler than against DDoS since the attack comes from a single source. DDoS on the other hand can be very complicated to avoid since the attack comes from many sources at once [21]. There is also the problem of a DDoS attack not actually being an attack (heavy loads on a news webpage after an impacting event could be classified as a DDoS attack) [25].

The upside of DoS attacks is that they are harmless in the sense that they don't steal or destroy sensitive information (presuming that backups are created of that data) [16 p.184].

4.2 Malicious code[16]

A virus is a piece of malicious code that uses a host program to execute and replicate itself into multiple copies as the host program is spread. Worms is self-replicating code, but uses a network as a medium to replicate itself over (viruses uses files as its medium). A trojan is some embedded code that uses a piece of software to be spread. Once a system is infected by a trojan it can start to do the things it was designed for, such as opening a backdoor into the system, monitoring behavior or just destroying data [25].

Although malicious code aimed at embedded systems is becoming more and more real (as late as two weeks before this section of the report was written an article on the cell phone virus Lasco.A was released [16]), no attacks of this kind aimed towards hardphones are yet known to the author of this report. Softphones, on the other hand, often running on an insecure OS (or a secure OS with insecure software) are very much a possible target for these kinds of attacks.

Malicious code is probably a thing to come for IP hardphones. As several hardphones contain preinstalled programs and support for protocols they clearly become a target for possible attacks. Even if any malicious code aimed at hardphones are yet unknown, this is truly something to have in mind when designing future systems.

Malicious code aimed at regular personal computers are beyond the scope (and would take probably take a considerable amount of text to analyze) of this report and will therefore not be discussed.

---

[14] The notion DoS also includes DDoS attacks for the rest of this document, unless an explicit separation between the two are made

[15] Sending, receiving, digitalizing, encrypting, decrypting, jitter compensation and more

[16] Malicious code in this document refers to trojans, viruses and worms.

4.3 Spam

According to the TopTenReview™ website Spam Filter Review, 40% of all emails received are considered to be spam. To put it in other numbers; that's 12.4 billion spam emails or 2,200 per person and year [14].

Stopping email spam is far from a perfected technique as these numbers impart. Spam in VoIP is a serious concern[17] and the problem has already been acknowledged by several industry leaders. SPIT works in the same fashion as email spam with the difference that it's a voice message instead of a body of text [13]. Every type of communication method available through VoIP, be it multimedia, instant messaging or regular voice conversations, seems to open up for a new way for spam to be received and delivered.

Of all the problems discussed in this paper, spam is the one that causes the least amount of damage, but on the other hand, by far the most annoying one. The thought of prerecorded spam messages sent to your home telephone makes most people faces twist into a sort of painful expression. So, what is there to do against this?

J. Rosenberg et al of the IETF [13] released an Internet draft that, while according to the authors being unsuitable as a reference[18], still contains very interesting information on the issue of SIP and spam. The draft discusses three different type of spam; Call spam, IM spam and presence spam.

The most interesting type of spam out of these three is call spam. Spam in textual form (such as mail spam and IM spam) are problems that has been around for some time and are not really of interest to describe in further detail in this document.

A multitude of different approaches for prevention of call spam are listed in the IETF draft, but most fail to deliver a good protection against these kinds of "attacks". The most popular type of protection against mail spam is content analysis. In this approach the mail is parsed and keywords that might give away the mail as spam are looked for. This mechanism is unfortunately unavailable in VoIP. Calls are most likely to be encrypted, and even if they are not, the filter would still have to perform speech recognition to be able to determine whether or not the call is spam.

Another measure of defense is whitelists and blacklists. These would correspond to allowed and disallowed numbers. The problems with these are that blacklists only marks one sender as spam at a time. This list can grow at a rapid rate, and yet still provide a good protection against spammers. A whitelist is probably the ultimate spam protection, as all senders not on the list are blocked. This, of course, means that only the persons you put on the list can call you. Imagine the frustration when you learned that some game-show tried to call you to award you their $5,000,000 grand prize, except that they weren't on your whitelist.

Another interesting approach against call spam is that of manual effort. To make a call to someone you haven't called before some sort of work is required. Since computational work could be done by the same system sending out spam, it is important that the work is something a computer would have big trouble doing. One suggestion is to redirect the call to a service that randomly generates a number, announcing these to the caller (with background music running) and make him repeat the numbers back. This would make automates spam mechanisms much harder to create. Doing this each time you make a call could of course prove to be exactly as annoying as spam is to receive.

---

[17] It even has its own acronym; SPIT

[18] This is due to the document being seen as a work in progress

The IETF draft deals only with spam in systems using the session initiation protocol, but the spam problems of VoIP are not only bound to systems supporting SIP. The measures of defense discussed above are likewise general solutions, not SIP specific.

4.4 Adware and Spyware

Adware is short for Advertising Software and does exactly that – advertises. These advertisements often come in the form o banners, but could as easily be in automatic redirects when browsing. If Adware is said to provide information (as useless as it may be), Spyware then takes information. A piece of software is classified as Spyware if it sends information (without consent from the user) to a third party [17].

As with the malicious code, Adware and Spyware still seems to be a possible thing to come as no known cases of this type of software running on IP hardphones are known to the author. And as softphones runs on platforms that already are the main target for Adware and Spyware, they also become vulnerable to these types of programs. One concern is that Spyware will start to harvest information about what calls you make instead of what webpages you visit [1].

The best way to avoid Ad- and Spyware is preferably only to buy software, instead of downloading it of the internet. The programs you do download should be publicly known as Adware and Spyware free. Also, make sure that a firewall is installed that controls access to the network, so that Spyware trying to send information through backchannels are blocked.

If you suspect that your computer is infected by Ad- or Spyware it's a good idea to get a removal tool, such as AdAware© [31] and Spybot© Search & Destroy [32].

4.5 Traffic interference

Interfering with traffic sent between two parties communicating via a network is often the first problem that comes to mind when discussing VoIP security. Integrity, authentication and privacy are some important key issues when it comes to VoIP [1].

Integrity means that the information sent is also the exact same as the information being received [1]. An example would be to intercept the packets sent as a user makes a bank transaction, making the money intended to be transferred between the customers own accounts to end up in the perpetrators Swiss bank account.

Authentication is necessary to prove that someone is who he really is [1]. In the previous example with the bank transaction, this would mean that the attacker didn't have to intercept any traffic but just figured out a way to impersonate the customer and make the bank think that it is in fact the owner of the account currently communicating with them.

Privacy (or secrecy) in packet based networks means that eavesdropping on information sent isn't possible [8]. If privacy is properly enforced, no one should be able to acquire the actual information sent. Translated into a scenario with verbal communication, total privacy would be to talk in a code language that only you and the recipient would understand. Any bystanders would not have a clue to what you were talking about.

Avoiding traffic interference is in strong relation to what protocol is used to transmit data. If the protocol is not secure, the data sent could be compromised. According to Jonathan Rosenberg [5], one of the founders of the SIP protocol, VoIP traffic using the SIP protocol can be completely secure when it comes to eavesdropping and traffic interference using the SRTP. SRTP is a profile for the Real-Time Transport Protocol giving it the

possibilities of confidentiality, message authentication and replay protection [23]. Unfortunately, these kinds of attacks still remain an issue due to the corporation's lack of demand for the SRTP protocol, according to Rosenberg.

Another way of making traffic between endpoints more secure is by using a Virtual Private Network[19]. IPSec is a widely used suite of protocols for encrypting data and implementing VPN:s [24].

As these solutions suggest, the problem with traffic interference seems not to be the techniques available, but rather the deployment of these within corporations and other subscribers of the VoIP technology.

4.6 Spoofing

Spoofing is to impersonate something else. A common example is IP-spoofing which in short means to fake a destination IP-address [18]. Various sorts of entities can be spoofed, such as participants, IP-telephones, IP-based servers, a router etc [3].

In the world of VoIP[20] this means possibilities for call hijacking, toll frauds etc. Hijacking makes calls end up at different destination than intended. This can be accomplished by spoofing an end-point or some routing device along the way to the destination. Toll fraud is the same as spoofing the identity of a caller, making the costs of your calls end up on his or hers bill [3]. Toll frauds can be accomplished through fooling various network entities that you are someone else or by just seizing access to the physical IP-telephone [19].

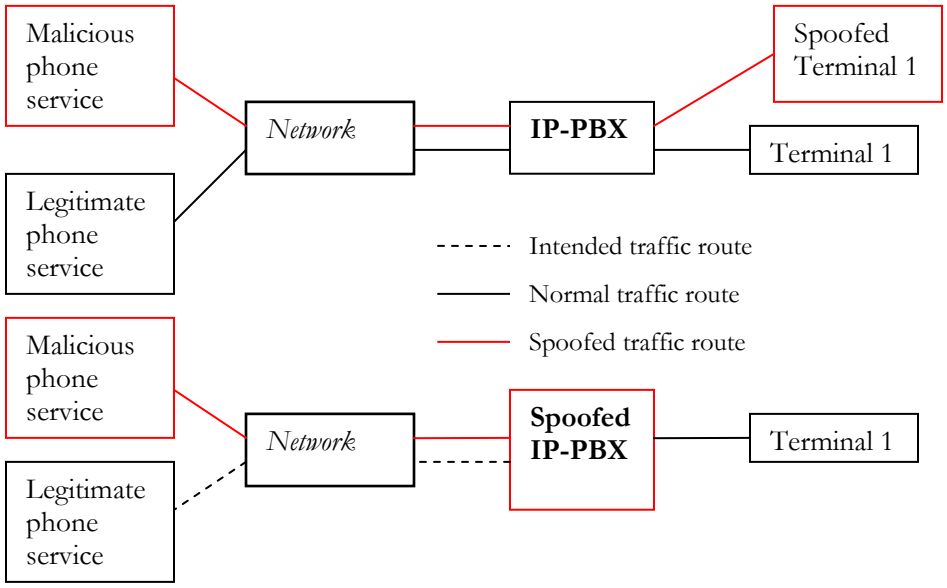

Figure 4. Overview of spoofing in a Voice over IP system. Top schema shows a terminal being spoofed and used to call a malicious service. The bill lands at the registered owner of the real terminal 1. Bottom schema shows a spoofed IP-PBX, redirecting a call to an unwanted malicious service.

[19] VPN
[20] Although not limited to that particular context

Although some actors on the market consider spoofing to be a serious threat to the market of VoIP [19], there is also documentation that spoofing can in fact be prevented by using the proper techniques, protocols and configuration. This is at least the statement made by IETF's[21] Jonathan Rosenberg in an interview made by eWeek© [5]. Rosenberg means that spoofing can be stopped by using a method called ingress source filtering. This technique in combination with SIP's built in functionality will be enough to make VoIP even more secure against spoofing than the PSTN. The reason for this is that besides spoofing a caller IP, the caller ID used to authenticate the caller also has to be spoofed. The PSTN is vulnerable to physical attacks on the phone lines, and can easily be spoofed by gaining that direct access. VoIP is meant to be dynamical, and no static connections are present. This means that even if you gain access directly to the SIP proxy, there will be know way to spoof another callers identity (unless you somehow got the caller ID from him or her).

4.7 Application layer attacks

Applications are never bug free, as shown in a recent study by Coverity®, discovering 985 bugs in the Linux source code [20]. Approximately two thirds of the bugs found in the study by Coverity® were in the critical part of the Linux kernel and were divided into the following groups:

- Crash causing
- Buffer overruns
- Performance degrading (resource leaking)
- Security

This shows that software bugs can be hazardous to the environment they are run on.

Application layer attacks not only applies to softphones[22] but also to hardphones. As mentioned in Chapter 4 most of the IP-telephones available on the market include some sort of software (such as TFTP or Telnet support). These pieces of software can contain bugs and in turn be a security issue in hardphones. One example of this is the flaws discovered in Cisco's® IP-telephone model 7960, that would lead to compromising of configuration files, MAC-address and much more [15].

Avoiding application layer attacks can be obtained by the use of proper guidelines when programming, intense testing and debugging and (once deployed) further external testing and proper patching and upgrading by system administrators [22]. A bug free environment seems yet to be an unreachable utopia. As with application layer attacks on hardphones, one can only wish that these techniques are used in full extent to avoid problems such as with the Cisco model 7960 mentioned earlier. It seems far fetched to imagine that software running on hardphones would never contain bugs or exploits.

[21] The Internet Engineering Task Force, official homepage at www.ietf.org
[22] This might seem as a bit of a paradox as a softphone, by definition, is a piece of software itself

4.8 VoIP security discussion

Security on the Internet has always been a huge thing. Corporations need ways to sell things online without revealing customers credit-card numbers, banks needed a way for people to manage their accounts online, libraries needed booklovers to be able to queue for books online etc. To be able to do all of these things over the Internet, there have been numerous solutions to the problem of authentication, data integrity and data secrecy. There is however one huge difference between security over the Internet and security over VoIP. PC users generally don't care about security unless they are doing something that they are familiar with as having to be secure, like bank transactions, purchases with credit cards and telephone conversations. Telephones are much more personal than computers (even more personal than the PC[23]) because you authenticate yourself with every word you say. Sending someone an IM message containing formatted text is like printing out a document and sending it by mail. There is no authentication involved other than more text; IP numbers and screen names. It's understandable why security in VoIP is the hot topic it has come to be.

Data secrecy and data integrity are things that can be solved with the same methods that are used in for example online banks. Authentication still needs some kind of secret key known only to the user. One solution would be to have the same kind of authentication methods used in online banking with for instance a card where you scratch out a new authentication code (like on a lottery ticket) every time you use their services. This is however not plausible since practically no one would accept the idea of going through that procedure every time you wanted to make a phone call. This calls for some kind of easier way of authentication, like a PIN code. Such a solution would make the phones more user friendly, but also more insecure when it comes to authentication.

The problem with data secrecy and anonymity is not only an issue for the general public, but also for government and emergency instances. Emergency operators would benefit hugely from being able to trace calls to locate people in emergencies, as would telephone operators to track down people maliciously using their services. Police and other government branches of security have certain needs to be able to taps phones and listen in on conversations between different parties. As packets takes different routes on the IP network, which is something that security benefits from, listening in on calls is made difficult. Even if the listener was given direct access to some key point in the infrastructure, the data would still be encrypted. This is a multi faceted problem that is still needs work.

One thing left untouched in the report but worth discussing is the cost of security. Securing gateways, developing and securing hardphones, implementing security protocols and keeping an eye on things is not free. VoIP is interesting only because it saves money, but perhaps the security upkeep for operators, manufacturers and subscribers could prove to be so expensive that the actual money saved by migrating to VoIP is lost through subscription fees, hardware purchasing cost, software maintenance and so on.

As seen in this report, many of the problems with VoIP can be avoided with the proper configuration, the appropriate protocols and good, well tested software. The two most problematic areas of security in VoIP are, not completely unexpected, spam and DoS attacks. They are both hard to stop, hard to track and hard to avoid. Spam has become a regular visitor of our lives, and a whole arsenal of email filtering programs exists to prevent

[23] Personal Computer – get it?

these irritating but often harmless mails from filling up our inboxes. In textual format, this spam can easily be overlooked, but if this plague spreads to the VoIP domain in the form of pre-recorded voice messages filling our voice mails or pestering subscribers in the middle of the night, then spam will become a problem in dire need of detection and elimination. DoS attacks are often aimed at strategic targets, such as web servers and gateways, with the aim of annihilating the quality of service for that particular target. As mentioned earlier in the report, VoIP is dependent on gateways, often running regular operating systems. These are the "perfect" targets for DoS attacks. A successful attack against one of these gateways could make phones unavailable for a long period of time. For the general public this would be annoying, and in the matter of emergency calls, deadly. Corporations could miss out on crucial information and miss deadlines. If gateways lack the proper protection against these kinds of attacks there would have to be some kind of backup system for making calls and that would contradict the whole idea with convergence. Costs related to keeping the backup systems up and running would be paid for as fees collected from subscribers, further decreasing the actual profit from migration.

New technology means new possibilities of making money. The VoIP technology attracts new actors to the market. Many of these actors are inexperienced when it comes to communication solutions and/or eager to get their hands on the big cash that this kind of technology could bring. Some of these actors could even be considered unserious and just plain greedy. The deployment of premature and poorly tested technology can not only hurt the security of subscribers, but also their faith in VoIP, counteracting the further development of the technology.

Not only does new communication technology attract possibly unserious actors, it also attracts the eyes of the underground community. There is a serious risk in considering today's threat against VoIP as severe as it gets. The more subscribers VoIP attracts, the more interest from malicious users it gains. These persons will always exists and no matter how skilled the designers of the systems are, there will always be people equally well at dissecting and exploit what they've built.

If VoIP is ever to become a world renowned standard, it has too make sure to be dynamic in its assessment against attackers and types of attacks. More and more software is bundled with hardphones and more and more functionality is required by the corporations as they strive to make more and more money by converging away from their old static solutions of communication. This new functionality brings new security issues, and as the security issues evolves, the methods for fighting the attacks has to be there evolving with it. The security of VoIP is not something of today, not even something of tomorrow, it is something of every day it is used.

5 Convergence in practice

As mentioned in Section 1.1, today's three biggest networks of telephone communication are the GSM net, the PSTN and the IP network. Two of these are already converged in a way that allows calls to be made across the borders of the network. These two are the GSM network and the PSTN. Calls from cellular phones can be made to home telephones and vice versa. This convergence would be unnecessary if there was a way to converge both the PSTN and the GSM network with the IP network.

5.1 Voice over IP via Bluetooth©[24]

In an attempt to explore just how receptive the technology available today is to convergence between different mediums of data transportation, an application linking two of these mediums together has been developed called VoIPBT. Although the application remains incomplete due to reasons further discussed in the Section 5.2.5, the programming provided information on the practical problems with convergence. The goal of the application was to route traffic off the GSM network used by cellular phones and other handheld devices and onto the IP network using a Bluetooth© link as a link to a regular PC. Outbound traffic would be sent from the handheld device, through the Bluetooth© link and be received and sent by an application running on the PC. Figure 3 gives a simplified overview of this process.

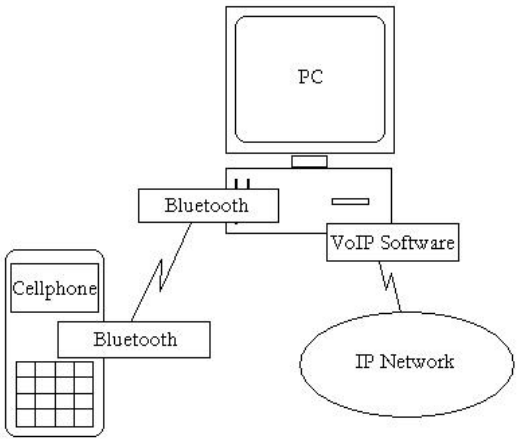

Figure 5. Overview of the functionality of VoIPBT

5.1.1 VoIPBT – Specifications

Prior to programming, an evaluation of Java (J2ME Wireless Toolkit 2.2) was made to see if it provided the functionality needed to complete the application. Evaluation was focused on the Bluetooth© API (JSR-82 technology) and also ways of accessing the microphone, speaker and call functionality. The evaluation of Java is further discussed in Section 5.2.4. The VoIPBT application was developed for Symbian© OS Version 7 in C++ using the UIQ 2.1 SDK. The choice of SDK was based on making the software supported by the Sony Ericsson 910i according to the document "Symbian OS System Definition" [33]. A third option consisting of writing the application for TSS own platform was considered, but never adopted or evaluated. Unfortunately this decision could have proven to "lock" the program to that platform, and make it less portable. Writing the application for TSS own platform could also prove to be difficult since some functionality might prove to be inaccessible.

[24] Further referred to as VoIPBT

5.1.2 VoIPBT – GUI Design

The GUI was created to be as user friendly as possible, while maintaining only the most basic functionality. This choice was made to cut down time put on implementing the GUI parts of the application. The GUI consists of three different views:

- Engine view
- Input view
- Dial view

This list corresponds to the order in which these views are presented to the user. The engine view provides the user with information on Bluetooth© initialization and setup. It also presents the selection dialog for the Bluetooth© discovery service. As soon as the initialization is done the view changes to the input view, which is responsible for receiving input from the user. The input view provides a simple textbox used for input of the telephone number that is to be called. This view provides the user with a bottom bar that has the buttons "Exit" and "Dial". When the user presses the "Dial" button, the number is parsed and checked for inconsistencies and then passed on to the dial view. This view is similar to the engine view as it only provides textual feedback on current progress. The dial view has a bottom bar with the button "Exit", which is used for both hanging up and exiting the program. If the user wants to make another call or abort the call (in case they dialed the wrong number for instance) he or she must use the "Exit" button to terminate the call and program and then restart it. The possibility of just hanging up and returning to the input view was discarded to save time implementing.

The toolbars and their respective buttons that are used in the program are loaded from what Symbian© calls resource files. The definitions of the toolbars and buttons reside in the `.rss` file in the program directory.

5.1.3 VoIPBT – System Design

The design of the VoIPBT system was entirely created with minimalism in mind. To make the project feasible over the amount of time given, functionality had to be cut down to a minimum and some aspects overseen. The most critical topics and some remarks concerning them are shown in Table 5.

Topic	Remarks
Memory management	Besides from handling pushing and popping of the parallel cleanup stack provided in the Symbian© OS development language (by using `CleanupStack::PushL( )` and `CleanupStack::PopAndDestroy( )`) in the initialization of objects (constructor and `NewL( )`), no memory management was included in the development. This decision was made to save time, but was, however, meant to be included in later implemental iterations.
Error handling	Error handling was never implemented during the development of the application.
PC application	For VoIPBT to be able to work correctly, an application to receive the data must be developed for the PC. This application was never intended to be a part of the master thesis project and was therefore completely left out. This application could possibly be made as a plug-in for some existing VoIP software to make transfer of data between the two easier.
Security	Security isn't something that necessarily needed to be dealt with as the program never was intended for commercial use. The only unsafe information would be that passing between the handheld device and the PC. This information would then have to be intercepted by someone in the same Bluetooth© piconet. This could be dangerous when dealing with secret information in an insecure environment, but this pilot application should on the other hand never be used in such a scenario.
Application performance	Application performance is closely related to memory management, especially in handheld devices with little memory and slow I/O. Performance could be not only be lowered by miss used memory management but also by using bad encoding algorithms (causing computational loads), poor program design and platform limitations. Performance was less important than getting functionality to work and was therefore left as a goal for later implemental iterations.
Network performance	Network performance is critical in real time applications such as VoIP programs. Due to the high bandwidth of Bluetooth© the delays taken in considerations were those from computations and overhead, not from network latency. The network performance of VoIPBT would be limited by the speed of the VoIP program used on the PC application and also by the type encoding used for the data sent and the attributes of the network data (packet size, packet structure etc).

Table 5. Interesting topics regarding VoIPBT

An overview of the program structure in the VoIPBT application is shown in Figure 4. The figure shows boxes representing classes and lines between them that describe the parent-child relation (read from top to bottom).

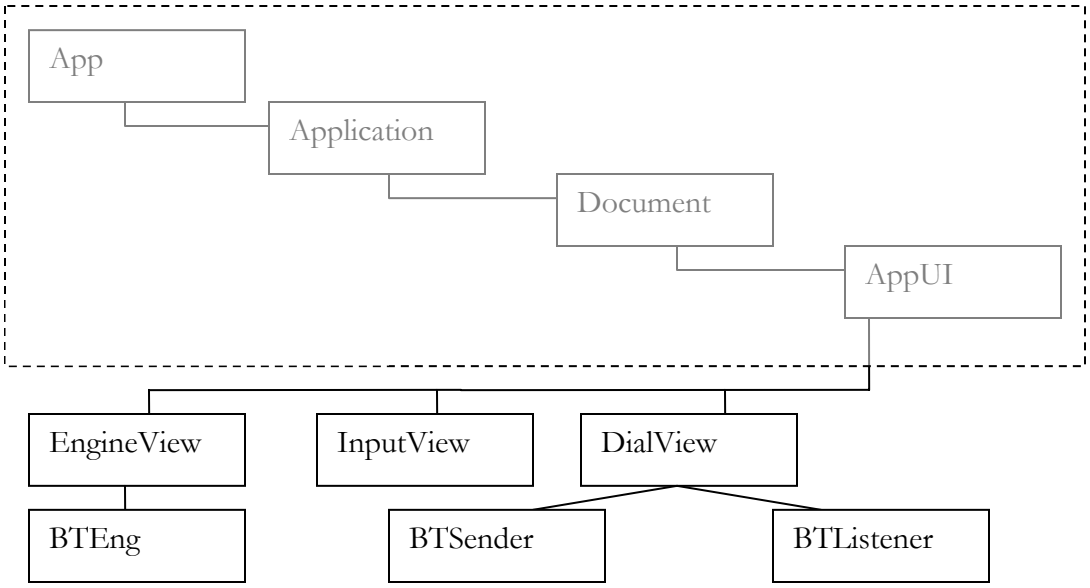

Figure 6. Structural overview of VoIPBT

To more closely specify the underlying functionality of the VoIPBT application, the following table lists some classes and a more thorough examination of them.

Class	Notes
`AppUI`	The `AppUI` class has been redesigned to handle all incoming commands from the different views, instead of each view handling their own commands. `AppUI` has also been extended with the `ChangeViewL(TViewType)` function for switching between views.
`EngineView`	The `EngineView` class handles interaction with the `BTEng` class. It calls the discovery functions in `BTEng` and handles the value it returns.
`InputView`	The `InputView` class handles the input of phone number to call from the user. This value is then passed on to the `DialView`.
`DialView`	The `DialView` handles the actual dialing, while starting the `BTListener` and `BTSender` objects and adding them to the scheduler. The `DialView` decodes and

`BTEng`	parses incoming data as well as formatting and encoding outgoing data. The `BTEng` class serves as the interface against the Bluetooth© devices that are used during the execution of the application. First it initializes the attached Bluetooth© device. Then it decides which type of Bluetooth© devices the application should include in the discovery process. Third, and finally, it launches the discovery service using the built in UI plug-in (with a call to `StartNotifierAndGetResponse( )` for a `RNotifier` object with the predefined `KDeviceSelectionNotifierUid` as a parameter). After the UI plug-in returns the device selected by the user, the result is passed on to the `EngineView`.
`BTSender/Listener`	As the manner of sending and receiving information to and from the device running the VoIPBT application could become delimited by the functionality present for third party telephony, no initial design for these classes was though of. The classes `BTSender` and `BTListener`, however, inherit basic functionality from the `CActive` class. The `CActive` (as in Active Object) class is Symbians© substitute for threads[25], and was considered a good start for asynchronous transfer of data.

Table 6. Class descriptions of VoIPBT

The VoIPBT application has one single flow of execution (excluding errors and premature terminations). A schema over the execution of VoIPBT is shown in Figure 5.

[25] Threads are, however, still possible to implement in Symbian

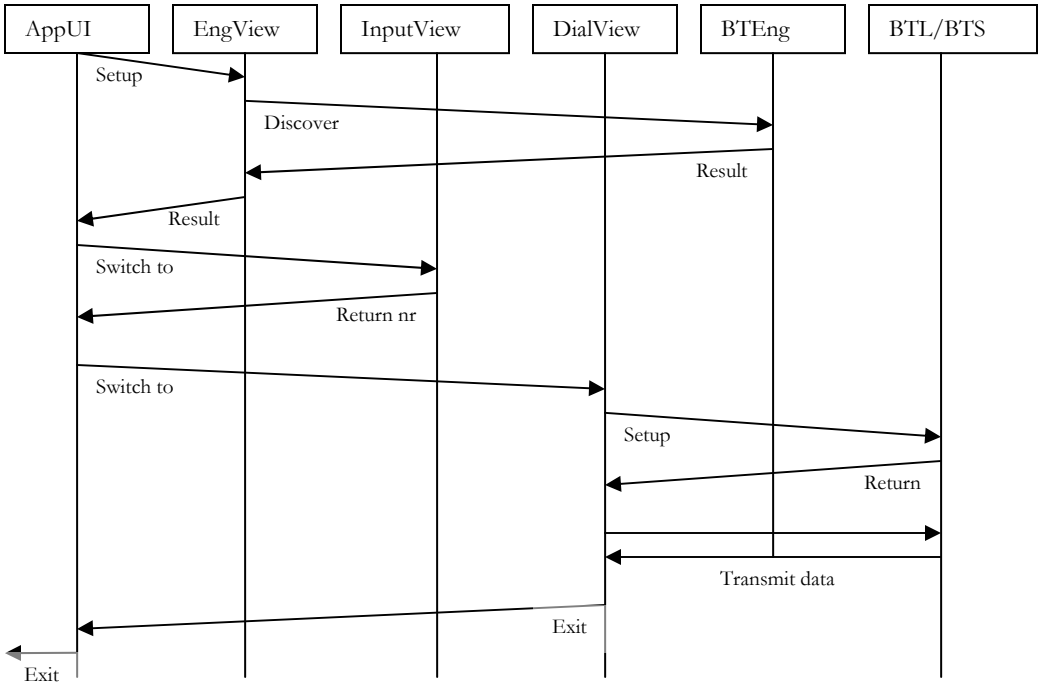

Figure 7. Flow of execution in VoIPBT[26]. The executions consist of the establishing of a call.

5.1.4 VoIPBT – Evaluation[27]

The big problem with coding VoIPBT, and this conclusion could be reached even before the actual coding began, is to gain access to the device microphone and speaker. Two different approaches on how to manage this were evaluated. The first one was to gain access to these resources by using some multimedia library to record through the microphone at the same time as using the speaker to send voice back to the user. The second was to use library call functionality to make the phone believe it was calling somewhere meaningful, while capturing the data sent and passing it on to the receiver.

The first approach was discarded due to various problems. Simultaneous recording and playback would probably be hard enough to manage, not to think of encoding and streaming of the data at the same time. There are also memory and delay issues with this approach. If the recording is to be streamed in real-time, there would have to be some mechanism that stops the recording, encodes the data recorded, sends if off and then deletes the data on the local device. All of this would have to be done after some predetermined size of data had been recorded, and the local device would have to start a new recording to capture what was being said during the processing of the last piece of data recorded. If this problem wasn't acknowledged, the recording could either lack large parts of the things spoken into the microphone in the case of no new recording starting while the last was being processed. The device could also run out of memory and cause large delays in transmission if only one recording per send was being done.

---

[26] BTListener and BTSender are renamed to BTL and BTS respectively in the Figure.

[27] Evaluation of the VoIPBT application will be based on initial design and not on functionality, as the development suffered premature termination.

The second approach, using some sort of platform functionality, was the one that was used for VoIPBT. The Symbian© C++ API provides an interesting functionality in the specification of the `CTelephony` class from the library `Etel3rdparty.lib` [34]. This class provides asynchronous functionality for establishing outgoing calls as well as waiting for incoming. As a call is established, an `RComm` object is passed along with it that is used to write and read data over the connection. This `RComm` object is in fact a serial port and uses an `RCommServ` object as a session with the serial communcation server on the local device. An initial investigation on this approach was conducted, but never fully evaluated due to the premature termination of the VoIPBT project. The conclusion made on these early investigations showed that the right module loaded by the `RCommServ` object a possible connection between the `RComm` object and regular sockets and further on, the Bluetooth© link could be established. When the `CTelephony` object tries to establish a call using `EstablishDataCall(RComm& ,const TDesC&)` it uses the `RComm` object passed to attach to the first data line supported by the local device hardware. It then tries to dial out on and pass data using that data line. Unfortunately, development never reached the phase of actual testing of this possible solution.

The reason why so much energy was put into making access to microphone and speaker work properly was to ensure that what was seen as the largest obstacle was already well thought through as the coding began. The knowledge on how to making the access to incoming and outgoing speech could also prove to play a role in how the rest of the application was designed and coded.

The Bluetooth© link that was required for the application to function was on the verge of completion. The built-in dialog for Bluetooth© discovery mentioned in Table 6 was completed but lacked certain functionality.

5.2 Further convergence

The VoIPBT application was created to test the possibility of conversion between the GSM network and the IP network through a Bluetooth© link. Not only was the applications functionality limited by the link between handheld and PC but also by programming language, platform and type of device it was running on. Changing these variables creates new ways of convergence. This section covers theoretical evaluations of some of these other solutions of convergence.

5.2.1 GSM, via IrDA, to IP

Another example that attempts to converge the GSM and IP networks are simply to change the link between the handheld device and the PC. A contender to the Bluetooth© technology is IrDA. IrDA (Infrared) can maintain bandwidths all the way up to 16Mbps which is more than enough to support both voice and video transfers without latency. The setback is of course the range of IrDA which is suggested from 0 to 1 meter with in a 30° cone [35]. Low range and high bandwidth is good for raw data transfers, but severely reduced the freedom of movement for the handheld device and its user. Making a cordless device dependent on distance from the PC would go against the original idea, making a handheld device act as a portable phone establishing calls via the IP network instead of via the PSTN.

5.2.2 GPRS to IP

As GPRS uses timeslots to send packet switched data, send and receive speed in GPRS is asynchronous and highly variable, which leads to some difficulties in assessing whether or not it would serve as a possible way of linking a handheld device together with a PC. The bandwidth in one direction could support good quality and no latency, and the other direction could become a bottleneck by only being capable of highly compressed of data without latency problems, if any at all. A test using TP Test [36] on a Sony Ericsson T630i gave the result of a download speed of 33.6 kbps and an upload speed of 10 kbps for GPRS. The test shouldn't be seen as a scientific study, but as an indicator on how slow GPRS connections can be and how much difference there could be between upload and download speeds. Voice codec's can encode voice into very low bandwidth requirements[28] and since IP headers can be added once the data is received at the PC, there is no need to include the extra IP bandwidth requirements, which is around 16kbps [37]. As the speed in the TP Test was measured to a low of 10kbps upload, there shouldn't be any problems keeping latency down on that particular connection. It is, however, a very close call.

According to Rob Barden of IFR Systems[29] [38], almost all GSM/GPRS cellular phones are of type 1, meaning that they are only half duplex, not being able to send and transmit data at the same time. This is of course a major obstacle when trying to use GPRS for transmitting speech and must be overcome to avoid having to push a button before talking (push-to-talk).

The duplex issue and the borderline case with bandwidth prove GPRS to be a bad choice for creating convergence between a handheld device and the IP network.

5.2.3 3G

3G is the next generation of mobile network designed for high-speed transmissions of data. Speed varies from 144Kbps up to 2Mbits depending on environment in which the device using the 3G network is located [39]. Unfortunately the third generation of mobile networks still has what VoIP is trying to solve in the PSTN; a static infrastructure. The 3G network requires special masts to be put up to handle the network. This makes it a completely new network, instead of incorporating one network into another (such as PSTN into IP). This new network will surely prove to be a valuable asset in the future of mobile communication, but still doesn't utilize any existing solution and therefore cannot be seen as a form of conversion.

5.2.4 Java

Besides from changing the link used to create conversion, the language for development can also be changed. This doesn't create a new type of conversion, but it alters the different variables (such as speed and stability) of the actual application. Besides from this, there is also an interest in seeing just how easy things are to develop in different languages. Development takes time and, yes, time is money.

One of the main goals with the highly versatile language Java was to create platform independence. Because of this, the language can be written on one platform and

[28] 5.6 kbps for G.723.1 MP-MLQ
[29] Official homepage at *www.ifrsys.com*

easily moved to another. This, however, makes Java very unsuitable for writing low level programs. The optimal language to write VoIPBT in would in fact have been Java since then the program could have med written as a Midlet for easy porting to several different platforms and phones, but it was established by both the author and other persons involved in Midlet development at TSS Umeå that Java wouldn't be able to provide the required functionality. More specifically, the decision not to write VoIPBT as a Java Midlet was decided because of the lacking support for microphone access and third party telephony.

For Midlets to work properly and to acquire more hardware close functionality, it needs to be signed. Certificates for signing are provided by different phone manufacturers. There are economical factors involved in getting certificates, and during the investigational phase of Java it was established that getting the proper signing would probably not only cost money but also be time consuming.

5.2.5 Conclusions

Information concerning the feasibility of making VoIPBT a reality is extremely scarce, and as far as the author knows, no other application that does this has been released. Several forums, for instance NewLC and Forum Nokia [41 & 42], and a newsgroup [43] were looked through to see if any similar work had been done, but many of the questions posted got minimal or no response. Creating convergence between networks using software has, as Section 5.2 shows, two sides to it. The first is the hardware. In order to establish some sort of connection between two networks, there needs to be some sort of hardware link. In Section 5.2 some different type of hardware was evaluated as to whether or not they could be used for creating conversion between different networks. The other side of creating convergence through software is the development environment. In VoIPBT the language was C++ and the application was compiled for the Symbian© OS platform. The development environment and the hardware each provide one crucial key to the success of the software. Using unsuitable hardware could increase limitations of the program and using a bad choice of language and/or platform for the software could cause the time of development to run through the roof, or even worse, to run into a dead end due to lacking functionality support.

6 Summarizing discussion

This report was written for a large audience and is width oriented. A lot of in depth information has, on purpose, been left out to make the report more understandable for people lacking major knowledge of computers. No other protocol than SIP is discussed in this report, which is a decision made through protocol analysis, market demand and personal preferential.

The market analysis in the report has been left short for one major reason. This master thesis project is written as the last part of studies in computer science. The focus of this report is therefore the technological aspects of VoIP (more specific convergence and security). The market analysis performed in this report is a direct request from the employers

at TSS Umeå. Further investigation of the market interest and money to be earned through migration to VoIP systems are left to people working in more suitable domains.

One problem when writing about VoIP security is the lifespan of the information published. As this is relatively new ground in the area of security, much information is unsuitable as reference material as it can change from day to day. The majority of information is bundled into articles and interviews. The quantity of scientific information regarding network security, on the other hand, is of course large. The author has tried to refrain from using sources of "regular" network security as much as possible.

Another unfortunate trend in the available material concerning VoIP security that is to be found on the internet is the influence of personal opinions in the text. First there is the cautious camp of writers[30] that are eager to highlight all the possible pitfalls of VoIP security. They often refer to current security issues and how these could affect the development of VoIP. The other camp of writers is the visionaries, that praise just about everything VoIP and tries to fight of all accusations of VoIP being insecure. These can often be representatives of different manufacturers and developers, or just people believing in VoIP. The problem grows as article after article finds it way on to the internet, the first one contradicting the second. The articles are often filled with personal reflections and lacking proper references. This not only affects reports such as this, but more importantly, affects the public view of VoIP security.

The VoIPBT application was an attempt of creating convergence between two different networks. Even though the application was never finished, much valuable information regarding the possibility of convergence was extracted from this attempt. First of all, the possibility of writing programs in certain programmatical languages for handheld devices is limited to signing from manufacturers. This process is to make sure the program isn't allowed to do attain permission to certain functionality. In other words, to stop the program from doing something the manufacturer hasn't given consent to. Secondly, there are still things to wish for when it comes to core functionality in different languages. C++ for Symbian© has a more open SDK than for instance Java, but it still maintains (irritating, yet reasonable) focus on more common tasks, such as UI development. Initiating and modifying calls is a tricky business, even on such an open platform as the Symbian© OS.

There are more obstacles in the path of VoIPs evolution besides from functional restrictions during development. VoIP could be a bandwidth consuming business if the demands for availability, QoS and sound quality are to be kept. The world is getting more and more connected to the internet, and more and more people are reachable just by using IP-to-IP telephony. The downside is that even though more and more people are getting attuned with the networked world, the bandwidth of most of these connections is slow. Modems are almost a part of the past in more developed countries (such as Sweden), but in many parts of the world they are the only reasonable way for people to connect to the internet. This problem has economical roots. Some countries just cannot afford the expenses of a fast network infrastructure and other countries, which have the money, just don't prioritize it high enough.

If VoIP is to be interlinked with the Internet, the technology has to be able to evolve along with its network. One thing currently on the makings is the transition from IPv4 to IPv6. This new system is both to increase security and also to free up some much needed address space for new IP addresses, as the old ones are running out. These kinds of changes are something that needs to be dealt with by the VoIP community to make sure that there

[30] Some might even call them realists.

aren't any problems with security and/or interoperability. Perhaps there are even larger obstacles to avoid on the horizon. What if the majority of calls start to pass through the IP networks instead of through the present PSTN. Dependence of availability, QoS and security is moved from the operators that serve the PSTN to the operators that serve the IP network. Either these new operators want to have their share of money for the work they put into making everything function properly or the subscribers will have no guarantees when it comes to the VoIP service. Perhaps these operators can cut down their own costs (and, by that, the subscriber's costs) by eliminating the PSTN, but since the IP network has several different operators, who can demand money from whom? In the end, economical factors are what control the market.

7 Future work

Throughout this report there have been indications that much work still needs to be done when it comes to security and convergence. Many questions remain unanswered and much information comes from articles promoting their respective authors personal opinions. The lack of proper information concerning VoIP security and the little work that has been done trying to create software to support the new convergence opens the field for people to work in this highly interesting area. There are both current topics that could be looked closer into and also work that could be done in the future.

7.1 Current topics

- The technology of the hardphones. This report wasn't meant to be a low-level technology extravaganza, so the actual hardware design of the various hardphones should be interesting to look at. Perhaps the design of these will give a clearer picture on the threat of malicious code etc in hardphones?
- Protocol evaluation. SIP is already somewhat of an industry standard, but how well is it designed? Is it really all that it claims to be? Can it live up to the expectations? Will it work equally as good in the future?
- Economical analysis. A more in depth analysis of the different economical factors playing in would be interesting. What actors are there on the local/global market? How much does it actually cost? Are there costs involved that are easily overlooked when calculating profit through migration?

7.2 Future topics

- The evolution of hardphones. Will hardphones become the new thing to integrate with a regular personal computer? As they are already hooked up to the same network, synchronization tasks would be easy to implement, but can exploitation of this connection be a security risk? What kind of protocols will hardphones support in the future? Will there be much more advanced programs? Will this perhaps introduce malicious code to the world of hardphones?

- Evaluation of security risks. How well do reports like this one depicture the reality? Do the threats seen as severe today really become such a big issue further on and will any issues that are not considered alarming at the time of this report become so in the future? Have any new threats emerged?
- The evolution of the VoIP. Will the IP network deployed today be the final carrier for the VoIP technology? Will a completely new, isolated network be deployed to minimize security risks and increase QoS? How will the airborne mobile network converge with the VoIP technology? How will all the calls passing through the network affect bandwidth?
- The economy of VoIP. Who will pay the bill for all the new traffic? Who will make sure that QoS is fulfilled? How much money is there to be made in the end? What will happen to the PSTN?

8 Closing comments

Voice over IP could be the next big thing. Revolutionizing the telecommunication and possibly eliminating an entire network in the making is not something that happens every day. A fortunate thing with voice over IP is that it brings possible earnings for corporations, something that is always welcomed by the market.

Not all that glimmer is gold is an expression commonly used, and awfully true in this scenario. Voice over IP obviously has a multitude of problems to solve and pitfalls to avoid before it can ensure its own place in IT-heaven. If the technology ever takes the world by storm is still to be seen, but it is probably best if you try not to hold your breath until then.

9 References

All references verified 18/5 -05.

[1] – **Sicker, Douglas C. & Lookabaugh, Tom**, *VoIP Security: Not an afterthough,* September 2004
<http://www.acmqueue.org/modules.php?name=Content&pa=showpage&pid=209&page=1>

[2] – **Klein, Alan**, *Security Analysis: Traditional Telephony and IP Telephony,* 2003,
<http://www.giac.org/practical/GSEC/Alan_Klein_GSEC.pdf>

[3] – **Arkin, Ofir**, *Why E.T. Can't Phone Home?*
Security Risk Factors with IP Telephony based Networks, November 2002,
<http://www.sys-security.com/archive/papers/Security_Risk_Factors_with_IP_Telephony_based_Networks.pdf>

[4] – **Louderback, Jim**, *Security holes make VoIP a risky business,* May 2004,
<http://www.eweek.com/article2/0,1759,1591129,00.asp>

[5] – **Muraskin, Ellen**, *A pioneer's view of VOIP and SIP security,* May 2004,
<http://www.eweek.com/article2/0,1759,1593991,00.asp>

[6] – **Cisco Systems**, *SIP: The next step in converged IP communications,* 2004, <
<http://www.sipforum.org/index.php?option=com_docman&task=docclick&Itemid=75&bid=13&limitstart=0&limit=10>

[7] – **Radvision Ltd.**, *SIP: Protocol Overview,* 2001, <
<http://www.sipforum.org/index.php?option=com_docman&task=docclick&Itemid=75&bid=11&limitstart=0&limit=10>

[8] – **Rosenberg, Jonathan et al**, *RFC 3261,* June 2002,
<http://www.ietf.org/rfc/rfc3261.txt>

[9] – **Rosenberg, Jonathan & Shockey, Richard**, *The Session Initiation Protocol (SIP): A Key Component for Internet Telephony,* June 2000,
<http://www.cs.columbia.edu/sip/articles/SIP_tutorial_CT_Magazine_June2000.pdf>

[10] – **CERT/CC**, *CERT/CC Statistics 1998-2005,* 2005,
<http://www.cert.org/stats/cert_stats.html>

[11] – **Cisco Systems**, *Cisco 7900 Series IP Phones,*
<http://www.cisco.com/en/US/products/hw/phones/ps379/index.html>

[12] – **Snom Technology**, *Snom 220,* 2003,
<http://www.snom.com/download/data_snom220e.pdf>

[13] – **Rosenberg, Jonathan et al**, *The Session Initiation Protocol (SIP) and Spam,* 2004,
<http://www.jdrosen.net/papers/draft-rosenberg-sipping-spam-01.txt>

[14] – **TopTenReviews Inc.**, *Spam statistics 2004,* 2004,
<http://www.spam-filter-review.toptenreviews.com/spam-statistics.html>

[15] – **Arkin, Ofir**, *The Trivial Cisco Phones Compromise,* September 2002,
<http://www.sys-security.com/archive/papers/The_Trivial_Cisco_IP_Phones_Compromise.pdf>

[16] – **Lemos, Robert**, *Cell phone virus turns up the heat,* January 2005,
<http://news.com.com/Cell+phone+virus+turns+up+the+heat/2100-7349_3-5520003.html>

[17] – **Adware.info**, *Quick Reference on Adware and Spyware Software,* 2005,
<http://www.adware.info>

[18] – **Tanase, Matthew**, *IP Spoofing: An Introduction,* March 2003,
<http://securityfocus.com/infocus/1674>

[19] – **Internet Security Systems Inc.**, *VoIP: The Evolving Solution and the Evolving Threat,* 2004,
<http://documents.iss.net/whitepapers/ISS_VoIP_White_paper.pdf>

[20] – **Coverity Inc.**, *[No title],* 2004,
<http://www.coverity.com/news/news_12_14_04_story4.html>

[21] – **Chau, Hang**, *Network Security – Defense Against DoS/DDoS Attacks,* September 2004,
<http://www.securitydocs.com/library/2576>

[22] – **Conry-Murray, Andrew**, *Application-Layer Protection,* July 2004,
<http://www.networkmagazine.com/shared/article/showArticle.jhtml?articleId=22103705>

[23] – **Baugher M. et al**, *RFC 3711,* 2004,
<http://www.ietf.org/rfc/rfc3711.txt>

[24] – **Internet.com**, *Webopedia: What is IPSec?,* May 2004,
<http://www.webopedia.com/TERM/I/IPsec.html>

-[14] **Kurose, James F. & Ross, Keith W.**, *Computer Networking: A top-down approach featuring the internet,* 2004, ISBN 0-321-22735-2 [p.565 (security)]

[25] **Scheier, Bruce**, *Secrets & lies: Digital security in a networked world,* 2000, ISBN 0-471-25311-1

[26] - **Skype Technologies**, *Skype,* 2005,
<http://www.skype.com/>

[27] - **Skype Technologies**, *SkypeOut rates,* 2005,
<http://www.skype.com/products/skypeout/rates/all_rates.html>

[28] **Telia**, *Prislista för företagstelefoni,* May 2004,
<http://www.telia.se/upload/upload_doc/Prislista_Foretagstelefoni.pdf>

[29] – **B2 Bredband AB**, *Prislista för Bredbandsbolaget telefoni,* June 2004,
<http://www.bredbandsbolaget.se/files/pdf/Telefoni_prislista_040617.pdf>

[30] – **Nemertes**, *Vendors Vary Wildly on Real Costs for VOIP Rollouts,* Novemeber 2004,
<http://www.nemertes.com/node/view/392>

[31] – **LavaSoft**, *LavaSoft: Protect your privacy,* 2003,
<http://www.lavasoft.com>

[32] – **M. Kolla, Patrick**, *SpyBot: Search & Destroy,* 2004,
<http://www.safer-networking.org/en/index.html>

[33] – **De Jode, Martin & Turfus, Colin**, *Symbian OS System Definition,* September 2004,
<http://www.symbian.com/developer/techlib/papers/SymbOS_def/symbian_os_sysdef.pdf>

[34] – **Symbian Ltd.**, *Symbian Developer Library: CTelephony,* 2002,
<http://www.symbian.com/developer/techlib/v70docs/SDL_v7.0/doc_source/reference/cpp/ThirdPartyTelephony/CTelephonyClass.html#%3a%3aCTelephony>

[35] – **Vrana, Greg**, *Untangling IrDA and Bluetooth,* September 2001,
<http://www.edn.com/article/CA159700.html>

[36] – **IT-kommisionen et al**, *Bandbreddstest TPTEST,* [No date],
<http://www.tptest.se>

[37] – **Voice over IP Calculator**, *VoIP Bandwidth,* July 2004,
<http://www.voip-calculator.com/bandwidth.html>

[38] – **Elektronik i Norden**, *Att testa GPRS-mobiler skiljer stort från GSM*, May 2003,
<http://www.edtnscandinavia.com/tek/showArticle.jhtml?articleID=19500317>
The article is translated to Swedish and shortened down.

[39] – **3GToday**, *3G Technology,* [No date],
<http://www.3gtoday.com/technology>

[40] **Butler Group**, *Communications Convergence: Evolving to a next generation IP-based network,* 2004, ISBN 1-904650-09-0

[41] – **NewLC**, *NewLC*, 2005,
<http://www.newlc.com>

[42] – **Nokia**, *Forum Nokia*, 2005,
<http://www.forum.nokia.com/main.html>

[43] – **Symbian Ltd.**, *Symbian: Developer Network*, 2005,
<news://publicnews.symbiandevnet.com/discussion.epoc.C%2B%2B>

Appendix A - Glossary

TSS – Short for Teleca Software Solutions.
VoIP – Short for Voice over IP. This is used in the text as the technology for sending voice and fax over a network supporting the internet protocol. VoIP is based on converting signals into streams of bits that then are split up into packets and sent via the network.
PSTN – Short for Public Switched Telephone Network.
QoS – Short for Quality of Service. In VoIP this could be things like keeping the latency down, reducing jitter and disturbance and also just making sure you can make calls.
SIP – Short for Session Initiation Protocol.
H.323 – A protocol standard published by the International Telecommunications Union Telecommunications Section (ITU-T)
VPN – Short for Virtual Private Network.
IPSec – Short for IP Security. A set of protocols that promotes security as the IP layer.
DoS – Short for Denial of Service.
DDoS – Short for Distributed Denial of Service.
SPIT – Short for Spam over Internet Telephony.
Hardphone – A hardware IP telephone, as for instance Cisco® IP Telephone model 7960.
Softphone – A software IP telephone, as for instance Skype©.
GSM – Short for Global System for Mobile Communication.