AETA
AUDIO SYSTEMS

AoIP Codecs and network security

Contents

1. Introduction

Like all devices which use IP networks, and especially Internet, AoIP codecs raise questions and issues about network security. One can roughly see two types of concerns :

- The operation of audio codecs may create breaches in the security of computer systems, or even induce attacks. That is commonly the main concern for the IT managers.
- The system protection procedures and resources may hamper or even block the operation of the codecs. This time the problem is for the users, or the persons who want to set up the transmission equipment.

Of course both issues can be met together… As for any use of Internet, the solution should provide a good compromise between safety and ease of operation.

Without pretending to be exhaustive, we examine here some resources and solutions that can be implemented, and discuss their relative complexity and efficiency.

Pre-requisite: the reader should have a basic knowledge on IP networks, features of routers and firewalls.

2. Specific features of audio codecs

2.1. Use of UDP and RTP

The audio stream transmission normally uses RTP/UDP. Except for very specific and proprietary implementations, that is the case both for direct links without signaling and for sessions set up using the SIP protocol.

UDP is preferred to TCP for real time media, but it brings some issues whenever address translation (NAT) takes place.

Concrete example: in a signaling phase a codecs announces to the remote device that it can receive an RTP/UDP stream on its IP address and a port *n*. In fact, due to the presence of a router with address translation, these data are only valid on the local area network of the codec. From the remote codec, the IP address is invalid and/or unreachable, and the port number is wrong. The stream cannot reach the codec.

One way to deal with this is using STUN (cf. further).

2.2. SIP

The SIP protocol is often used by audio codecs, and it is required by the "N/ACIP" recommendation (EBU Tech3326).

SIP (Session Initiation Protocol) deals with signalization for a media session over an IP network between two "agents" such as audio codecs, VoIP telephones, etc. The signaling covers the setting up and releasing, with negotiation (using the SDP protocol) for the needed parameters: coding algorithm, IP addresses and ports.

The outline of a session progress is the following:

- The session initiator sends a request (INVITE message) to the remote "SIP agent".
- It proposes session setup(s) (using the SDP protocol), including coding, port numbers for the RTP stream, etc.
- The remote device accepts and indicates the selected setup.
- The agents exchange their audio streams with the parameters that have just been negotiated.
- At the end, one of the agents releases the session (BYE message).

The above describes a "peer to peer" link between the agents, and the SIP protocol does not impose the usage of a server. The SIP protocol normally uses UDP.

In a more advanced system, a SIP server will intervene in the signaling process. In such case the agents are first registered on the server before setting up sessions. Such a system brings several advantages:

- Insensitivity to changing IP address of codecs, usage just as simple as for common telephone systems
- Advanced features of server, beyond the essential proxy/registrar
- Better access/authorization control over the agents
- Possibility to tightly couple with firewalls

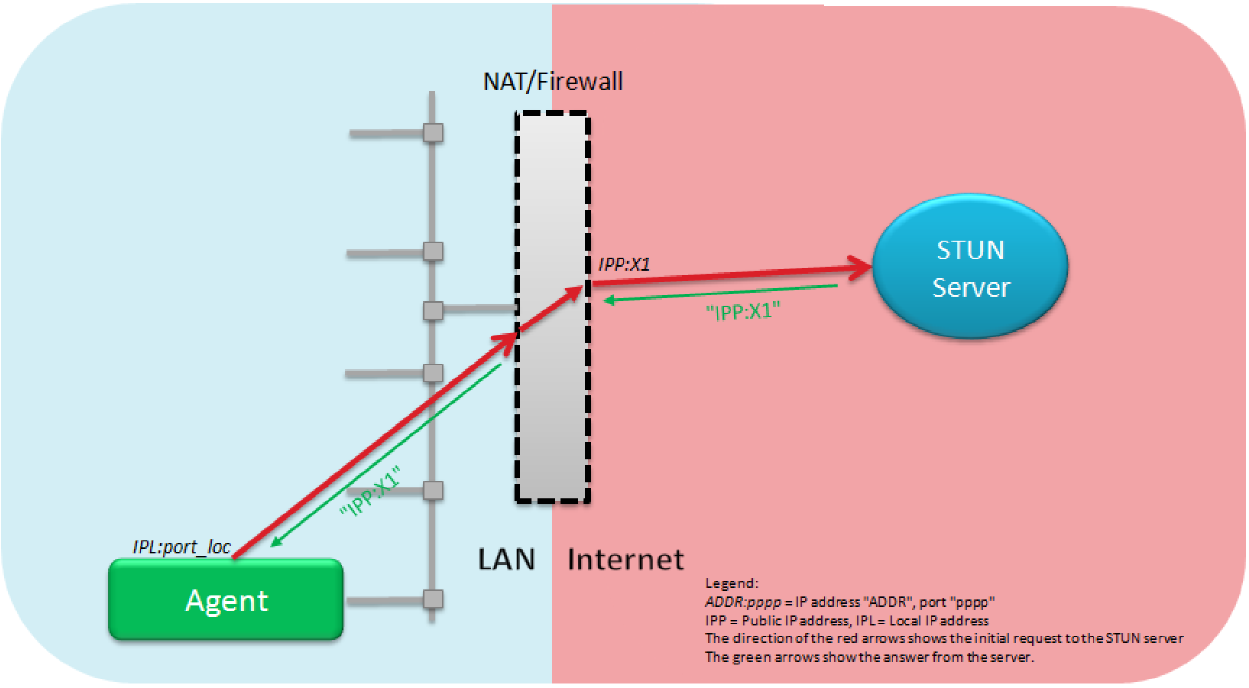
⇨ *The AETA codecs always use the SIP protocol, involving or not a SIP server.*

⇨ *Besides, AETA propose to its customers the services of a dedicated SIP proxy/registrar dedicated to audio over IP broadcast applications (follow this link for more information).*

2.3. STUN

The STUN protocol is a means to pass through routers with address translation (NAT). It allows the agents to discover the actual public IP address on the Internet side of the router, as well as the actual port number for the RTP stream.

Basic principle: the agent queries a STUN server, located in the public area, which sends back the correct information on the public IP address and the port numbers actually allocated by the NAT router. Then the agent can use these data for the signaling of a SIP session.


NAT/Firewall
IPP:X1
"IPP:X1"
STUN Server
"IPP:X1"
IPL:port_loc
Agent
LAN
Internet
Legend:
ADDR:pppp = IP address "ADDR", port "pppp"
IPP = Public IP address, IPL = Local IP address
The direction of the red arrows shows the initial request to the STUN server
The green arrows show the answer from the server.

⇨ *AETA codecs all provide the capability to use STUN. However this feature can be quickly disabled if needed.*

2.4. Other protocols

Codecs can also use other protocols for other needs than the essential audio transmission function.

- Remote control, usually under TCP/IP. Example: embedded html server.
- Monitoring, with TCP/IP or UDP/IP.

These features are secondary, so the related protocols and ports are usually optional. Therefore opening the necessary ports on firewalls is not mandatory.

3. Examples of vulnerability

3.1. *Direct attack on a codec*

A hacker may attack a codec directly on its control interface. The goal may be e.g.:

- Attempt to take control over the codec, opening the way to other malicious actions.
- Attempt to hang or crash the codec.

3.2. *Undesired connections*

Purpose: make the codec busy and impede its legitimate use at the appropriate time, or worse, disrupt the on-air program. Sometimes this can be done without even any illegal action or “hacking”, one just has to call a codec. In fact such trouble can be brought just as simply as an undesired telephone call… Even an operation mistake can lead to a similar undesired call issue.

In a more organized way, the attack may be carried out as repeated automated calls that overload the system or block it; that is quite similar to a DoS attack.

3.3. *Intrusion on the server*

The attack can aim at the server, trying to register in order to use illegally (and for free) its services, or to set up undesired links (which brings to the previous case). The hackers use specific tools in order to detect passwords or force registration.

A hacker may also try to intrude the server to get control over it, etc.

3.4. *Intrusion in the local network via the codec*

Penetrating the codec (via its control interface) is a possible way to intrude the local network, or other machines, like intruding a computer on the network (via a Trojan for instance).

However, an audio codec is not a target as classical as a Windows computer for example, unless it *is* in fact such a computer… But similar attacks on VoIP phones could be seen already.

3.5. *Session spying*

Audio codecs, like VoIP phones, may be eavesdropped for various purposes:

- In a classical way, for listening to the conversation. For a broadcast application, this may not be a big issue.
- For disrupting the link, or identity stealing.
- Getting information allowing to intrude the server, or other malevolent action.

3.6. Example case: SIPVicious

SIPVicious is an example frequently involved in malicious actions on VoIP agents, and it can be related to the various cases described above. In fact this is originally a set of Open Source tools, designed for testing and auditing SIP based VoIP systems. These tools have been diverted from their purpose by hackers who use it for:

- Detecting/enumerating SIP servers or phones (scanner)
- Trying to crack passwords that protect the SIP credentials
- Setting up pirate links

On a typical scenario, a hacker first detects the presence of a codec at a given IP address. Possibly assuming this is a SIP server, he/she then tries to set up VoIP calls via the device to telephone numbers. In fact this can't work, because the device should be a SIP server or proxy to deal with these calls. The codec « rings », receiving an undesired call from an unknown or unidentified caller, picks up the call. But it cannot get synchronized, and the line is released after a few seconds.

These « ghost » calls happen to repeat at a high rate, capable to lead to hanging and/or rebooting the codec.

For more about this topic, you can check e.g. this site: http://blog.sipvicious.org .

4. Problems induced by routers and firewalls

4.1. General

To describe it simply, the function of a firewall is to prevent, at the border between the local network and the Internet access, the transfer of undesired data, prone to create trouble. The firewall function is often integrated with the Internet access router.

One should be aware that, by its very principle, a firewall, whose purpose is to protect from malicious action, can as well impede legitimate use, if it is configured mistakenly or inadequately, or overzealously.

One other important point to keep in mind: beyond the network domain that is under the responsibility of the codec user, the network area under control of the operator may as well feature filtering and blocking rules. That is especially true for networks such as mobile data networks.

4.2. Protocol blocking

A firewall may happen to block data depending on the protocol used in the transmitted or received packets, for instance:

- Block UDP as a general rule
- Block VoIP signaling: SIP protocol. Similarly, a blocking rule may impact the STUN protocol.
- Block RTP streams. As an example, some mobile networks act just at this level to prevent the usage of VoIP in the network, even without necessarily intervene in the SIP signaling phases.

4.3. Port blocking

Instead of detecting and blocking a protocol, some rules aim at specific ports. The principle is generally to block any traffic, except the ports specifically enabled that will be “opened” to the traffic.

Conversely, the rule may be to target specific ports and block a given type of traffic. This is sometimes met in the operator network infrastructure, e.g. in order to reserve a type of service [to the operator], or in the Internet access router when it is provided and managed by the operator. The standard SIP and RTP ports are sometimes targeted by such rules.

This method is not always relevant, because it relies on the assumed fixed correspondence between a type of service and a port number. But for instance in a SIP session the RTP ports are negotiable, their numbers are dynamic. Open ports on a static scheme will not always work…

4.4. NAT routers

Here we refer to routers carrying out network address translation (NAT). Their primary function is to share a small number of public IP addresses (usually only one) among a rather high number of hosts on a private local network. The function of NAT is for this reason almost always included in an Internet access router.

The most frequent case is address translation with port translation (PAT), and this is our default hypothesis[1] all over this document when we mention NAT.

A more detailed and accurate description of NAT can be found for instance on this page:

http://en.wikipedia.org/wiki/Network_address_translation .

NAT routers happen to bring also a basic protection against external assaults, because unsolicited data cannot *a priori* enter the local network. Normally a "return" path is opened across the router when an outbound link is initiated, and the response packets use this path to reach the device that initiated the exchange and expect an answer. The path is closed again after some time of inactivity on this temporary link. Therefore an external source cannot take the initiative and send data to any device inside the local network.

With UDP and SIP, a NAT router brings specific issues for traversing it:

- Initiating a link from the outside is not normally possible, at least on a "peer to peer" scheme.
- During the SIP negotiation, the agents show the IP address and port they can receive the audio stream on. But the agent behind the NAT (aka "NATted") only knows and shows its local IP address and port, while the remote codec should know their public counterparts (as seen from the Internet side of the router). Hence the stream will not reach its destination.

Fortunately there are methods to overcome this issue, as for instance the STUN protocol, mentioned before in 2.3.

[1] However the issues and the implemented solutions are not much different in the other situations.

5. Access methods

We are describing here some methods that can be used in order to install codecs on a site and make it possible to set up audio links between a codec on site and another codec on a remote site. We assume that the remote unit is reachable via Internet, because a connection via a long distance private link is typically equivalent to a LAN connection, which does not raise specific security issues.

5.1. Direct connection to Internet

5.1.1. Principle

That is the most straightforward technique: place the codec directly on the Internet, with a public IP address dedicated to it. The operation is very simple and close to that on a LAN: the codec can directly create a link to the public IP address of another codec, or it can be reached from such codec by its IP address.

5.1.2. Advantages

- Very easy to set up. Configuring the access router may be a little tricky but this has to be done just once.
- No connectivity issue, i.e. no blocking to expect, because there is no protection…
- As the codec is isolated from a LAN, obviously it cannot propagate possible security issues to other units.

5.1.3. Drawbacks

- It is hard to imagine a worse solution for the codec security: it just has **no protection at all** against an attack or malicious use!
  The codec is prone to attempts to intrude it or take control of it, to undesired connections, eavesdropping, etc.
- The public IP address is imposed by the operator network; on some ADSL networks it is dynamic (not necessarily fixed over time). This makes it more complicated to set up links from remote.
- The codec is out of a LAN, which can be an obstacle to links with device inside the LAN.

5.1.4. Recommendations

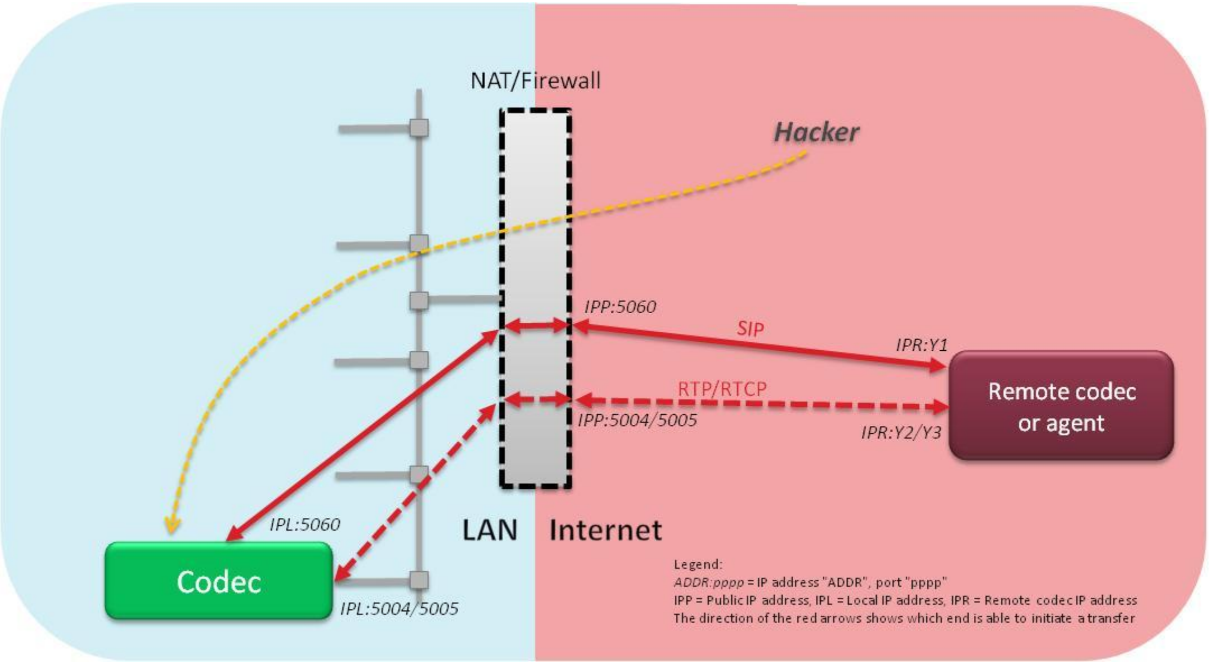
There is just one single recommendation: do not use this technique!

It is well known that a device dropped in the wild Internet like this is quickly detected and attacked, just a matter of minutes… There is little chance one can operate a device in such situation for a long time.

5.2. *NAT + DMZ*

5.2.1. Principle

The codec is, on a conventional scheme, installed behind an access router with NAT, but placed in "DMZ": it directly receives every packet arriving at the public IP address[1] of the Internet access, with possibly some exceptions (for example some ports may be kept reserved for the router or another piece of network equipment). Nevertheless it is on the local network and its IP address has a local scope, regardless if it is allocated statically or by DHCP.


NAT/Firewall
Hacker
IPP:5060
SIP
IPR:Y1
RTP/RTCP
IPP:5004/5005
IPR:Y2/Y3
Remote codec or agent
IPL:5060
Codec
IPL:5004/5005
LAN
Internet
Legend:
ADDR:pppp = IP address "ADDR", port "pppp"
IPP = Public IP address, IPL = Local IP address, IPR = Remote codec IP address
The direction of the red arrows shows which end is able to initiate a transfer

As the codec is not initially aware of its public IP address, the NAT issue is raised, as mentioned before in 4.4. A STUN server (see in 2.3) should be used so that the unit discovers its public IP (*server not shown here to simplify the drawing*).

5.2.2. Advantages

- Rather easy implementation, nearly as simple as the previous option (direct connection to Internet).
- No connectivity issues, provided that STUN is used. The codec can directly set a link to the public IP address of another codec, or it can be reached via its public IP address.
- Usually the codec can communicate with other codecs located on the same LAN.

[1] Or one of the public addresses, if the access includes several IP addresses.

5.2.3. Drawbacks

- Regarding the operational security of the codec, this system is just as dangerous as the previous one; there is just **no protection at all** against an attack or malicious use!
  The codec is prone to attempts to intrude it or take control of it, to undesired connections, eavesdropping, etc.
- The codec is not necessarily isolated from the LAN: in the worst case it can be used as a Trojan and propagate attacks.
- Some consumer routers may not allow such configuration. In any case the configuration is not always simple.
- The public IP address is imposed by the operator network; on some ADSL networks it is dynamic (not necessarily fixed over time). This makes it more complicated to set up links from remote.

5.2.4. Recommendations and alternatives

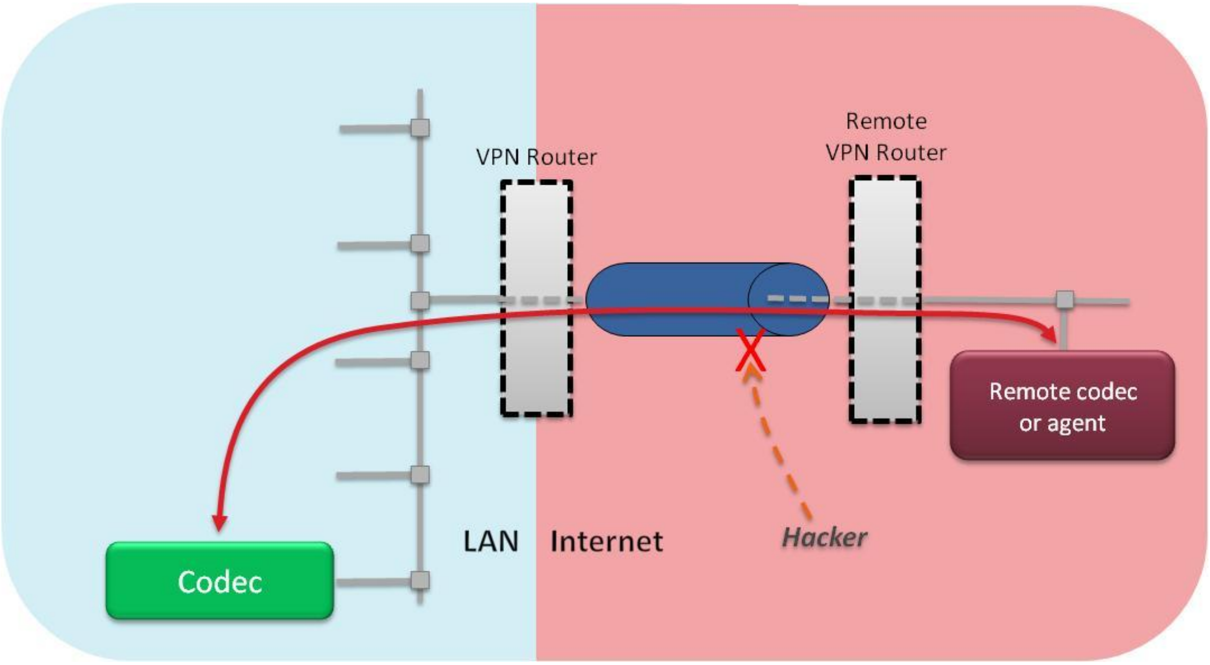
This set up is not recommended for the same reasons as the previous one: lack of protection against external attacks.

An alternative consists in isolating the codec in DMZ from the rest of the LAN, e.g. with a second firewall between the LAN and the codec. Such architecture is more complex but it has no positive impact on the vulnerability of the codec.

5.3. *Link via VPN*

5.3.1. Principle

Many users turn to a system radically different from the previous one: the codec is on a LAN, “hidden” behind a VPN router. This router will then, as its name tells, create a virtual private link with another LAN, equipped also with a VPN router. The other LAN is thus an extension of the private IP network. This is so called a “tunnel”: the *real* link between the routers is secured and encrypted, it wraps, like a hermetic and opaque tube, the data exchanged between the two sites.



5.3.2. Advantages

- Maximal security. A very high protection level can be ensured, and it is extremely difficult to intrude or snoop.
- The device is protected against any solicitation from outside the organization.
- The operation is as simple as on the local network, and in effect it is kept inside a private network.
- Such configuration fits all types of codecs, whatever their protocols, settings and fatures can be.

5.3.3. Drawbacks

- The initial setup is rather tricky, if not complex. This is commonly under control of the network manager, and is hardly possible without prior planning.
- Only those remote sites that are integrated in the private network are connectable. It is not possible to set up a link at short notice with a device on another remote site[1].
- The VPN may bring an increase in the external traffic, compared to the net traffic induced by the codecs, and also possibly an increase in the latency. These impacts depend on the kind of protocol used.
- When the network link is of poor quality, the VPN happens to lose sync sometimes. It takes a while to restore it each time, which may amount to seconds. Once again, this depnds on the protocol and/or VPN equipment.
  This issue is relevant for a link via a mobile network, which cannot be 100% free from occasional interruptions. The VPN may increase this problem, in the worst case, by translating such short interruption into several seconds of break.

5.3.4. Recommendations

Thi solution is very interesting, especially for securing regular links between well defined and stable sites, and if there is no need for an opening to the outside of the organization.

If one side of the links at least incurs frequent transmission errors, it is recommended to assess the stability of the VPN when facing these errors, and check they won't induce too audible interruptions. The impact on latency too should be checked, depending on the operational requirements.

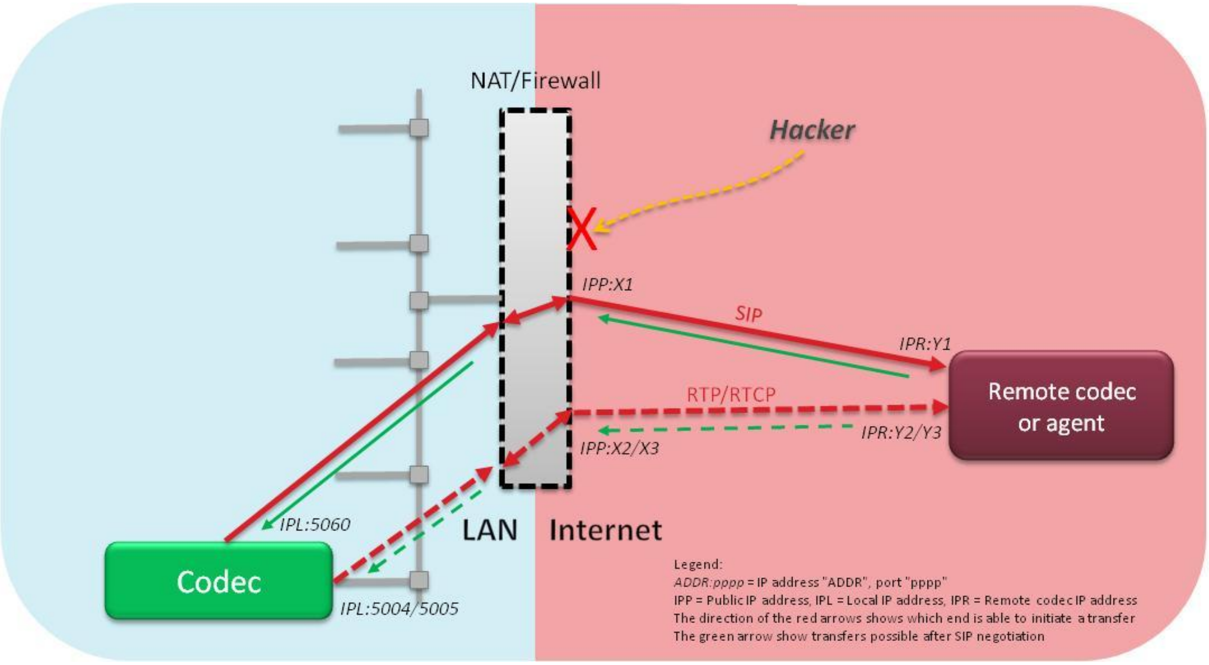
[1] Note : although this is paradoxical somehow, using a router with mobile network access is not considered here "at short notice", because then it is only a matter of activating/reactivating a link, prepared beforehand (no change in the equipment and the mobile network).

5.4. *“Standard” NAT*

5.4.1. Principle

The codec is, on a conventional scheme, installed behind an access router with NAT. This router often also acts as firewall to protect against external attacks.

As the codec is not initially aware of its public IP address, the NAT issue is raised, as mentioned before in 4.4. A STUN server (see in 2.3) should be used so that the unit discovers its public IP (*server not shown here to simplify the drawing*).



Legend:
ADDR:pppp = IP address "ADDR", port "pppp"
IPP = Public IP address, IPL = Local IP address, IPR = Remote codec IP address
The direction of the red arrows shows which end is able to initiate a transfer
The green arrow show transfers possible after SIP negotiation

The above diagram is valid for a codec implementing the SIP protocol. The packets which should go through the access router divide in two types:

- Signaling messages, using the SIP/UDP protocol.
- Compressed audio streams with RTP/UDP protocol.

When the codec sends a packet to the remote unit, or a STUN server, a path (and corresponding port) across the NAT router is opened. The feedback packets must use this path in reverse direction in order to reach the codec. Thanks to STUN the codec can inform its remote counterpart on the right address and ports, as seen from the public (Internet) side of the router.

In short, the codec can initiate a link and get the appropriate data to set up the link. In contrast, a session cannot be started on an external initiative.

5.4.2. Advantages

- The configuration is easy and does not imply a specific configuration for the NAT router. However, one must make sure to use STUN in the codec (or a similar method).
- It is not necessary to install or have available an infrastructure server, such as a SIP server. A STUN server must be used, though, and by its very principle it must be located on the public side. But this kind of server is not critical and many servers are available for free, for instance the one operated by AETA (stun.aeta.com).
- The codec can also communicate with other codecs on the same LAN.
- The system is rather well protected against external assaults, because the outside cannot simply initiate an exchange[1].

5.4.3. Drawbacks

- The codec cannot be contacted by a remote codec. This can be a major drawback or not, depending on the desired operating mode.
- Specifically, it is impossible to set up any link with a remote unit which is in a similar arrangement (behind NAT).
- For the specific case of routers with “symmetric” or “restrictive” NAT, this system does not work, because the ports discovered by using STUN are not valid for the subsequent transfer with the remote codec.

[1] However it remains possible to detect the public port numbers (dynamically allocated) during the legitimate transactions, and try to intrude through these access points.

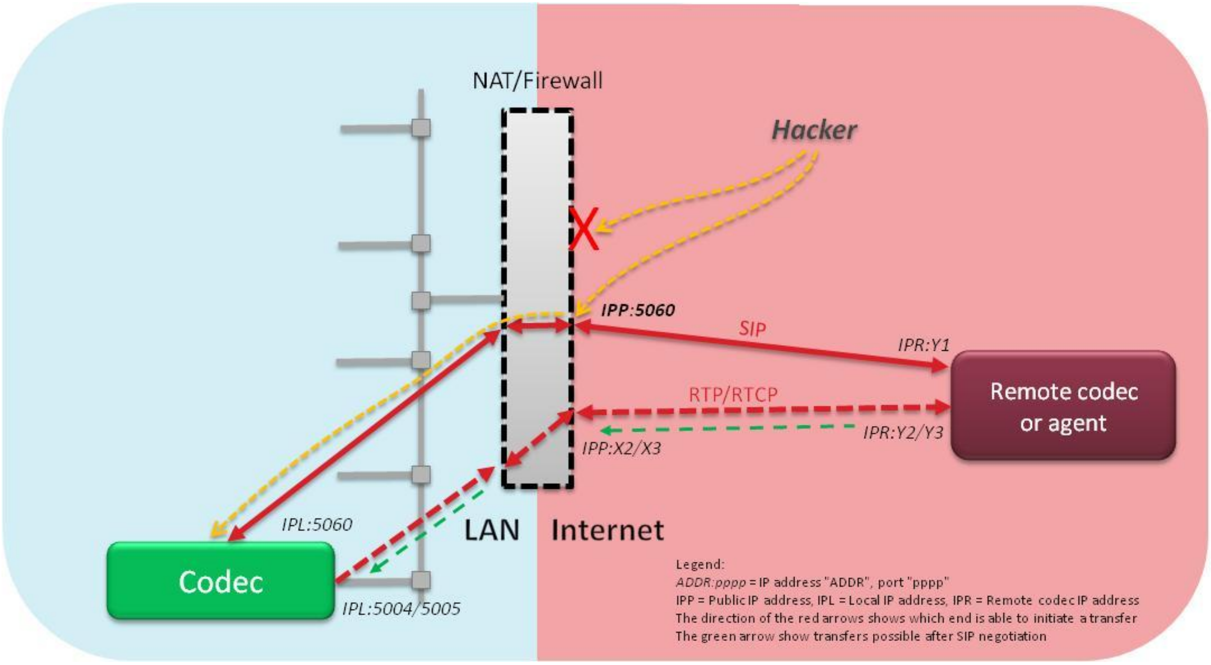
5.5. *NAT and port forwarding*

5.5.1. Principle

As in the previous situation (“standard” NAT), the codec is installed behind an access router with NAT. This router often also acts as firewall to protect against external attacks.

In order to provide a remote unit the capability to initiate a connection, a *port forwarding* rule is added in the router/firewall: the SIP/UDP port 5060 is forwarded to the codec.

Like in the previous case and for the same reasons, a STUN server must be used so that the codec gets its public IP address and port numbers (*server not shown here to simplify the diagram*).


NAT/Firewall
Hacker
IPP:5060
SIP
IPR:Y1
RTP/RTCP
Remote codec or agent
IPR:Y2/Y3
IPP:X2/X3
IPL:5060
LAN
Internet
Codec
IPL:5004/5005
Legend:
ADDR:pppp = IP address "ADDR", port "pppp"
IPP = Public IP address, IPL = Local IP address, IPR = Remote codec IP address
The direction of the red arrows shows which end is able to initiate a transfer
The green arrow show transfers possible after SIP negotiation

The above diagram is valid for a codec implementing the SIP protocol. The packets which should go through the access router divide in two types:

- Signaling messages, using the SIP/UDP protocol.
- Compressed audio streams with RTP/UDP protocol.

When the codec sends a packet to the remote unit, or a STUN server, a path (and corresponding port) across the NAT router is opened. The feedback packets must use this path in reverse direction in order to reach the codec. Thanks to STUN the codec can inform its remote counterpart on the right address and ports, as seen from the public (Internet) side of the router.

When it is the remote codec that initiates the call, it takes advantage of the static allocation of the SIP port to the codec. Afterwards, in the SIP/SDP negotiation, the codecs exchange with each other the needed data for setting up the RTP streams, including the port numbers.

5.5.2. Advantages

- Contrary to the previous case, here it is possible for a remote codec to contact the codec and set up a link.
- In comparison with the most basic solutions (see before *Direct connection to Internet* and *NAT + DMZ*), the codec is not exposed beyond the strict minimum needs.
- It is not necessary to install or have available an infrastructure server, such as a SIP server. A STUN server must be used, though, and by its very principle it must be located on the public side. But this kind of server is not critical and many servers are available for free, for instance the one operated by AETA (stun.aeta.com).
- The codec can also communicate with other codecs on the same LAN.

5.5.3. Drawbacks

- Although it is not very complex, the router configuration is slightly less simple as in the previous case.
- Sometimes it is not possible to apply this solution, either on too basic (consumer) routers, or by lack of access to their configuration. One obvious and unworkable case is the access via a mobile network: only the operator has control over the NAT access router to the network.
  As a consequence, this method is almost unusable for a mobile network access.
- With this method it is not possible to install more than one codec on a network access, unless several public addresses are available (one workaround for this issue is described further in the alternatives).
- The public IP address is imposed by the operator network; on some networks it is dynamic (not necessarily fixed over time). This makes it more complicated to set up links from remote.
- For the specific case of routers with “symmetric” or “restrictive” NAT, this system does not work, because the ports discovered by using STUN are not valid for the subsequent transfer with the remote codec.
- The capability for a remote codec to call in is available as well for any device or entity outside the organization, including undesired ones (see 3.2, Undesired connections)! **Therefore this system creates a breach, offering the outside world an uncontrolled access.**
- Possible security improvements involving the firewall (see further) are rather complex to implement and maintain.

5.5.4. Recommendations and alternatives

This system has many drawbacks, especially regarding security.

To reduce them, first it should be avoided to forward or open ports that are not necessary. For example, forwarding TCP port 80 (http) is a useless risk, unless it is actually necessary to provide a remote access to the codec's embedded html server.

⇨ *On AETA codecs, make sure to set up a password for the html pages access control, if you decide to open this access from outside!*

Conversely, the possible firewall should not prevent the codec to do the following when it initiates a session:

- Send SIP/UDP packets from port 5060 of the codec to the remote codec's SIP port (usually 5060).
- Send RTP and RTCP/UDP packets from ports 5004/5005 of the codec to the remote codec's RTP/RTCP ports (usually 5004/5005).
- Send requests to the STUN server (port 3478).

⇨ *The above listed port numbers are the standard values normally found with AETA codecs. On these units it is possible to change these ports at will.*

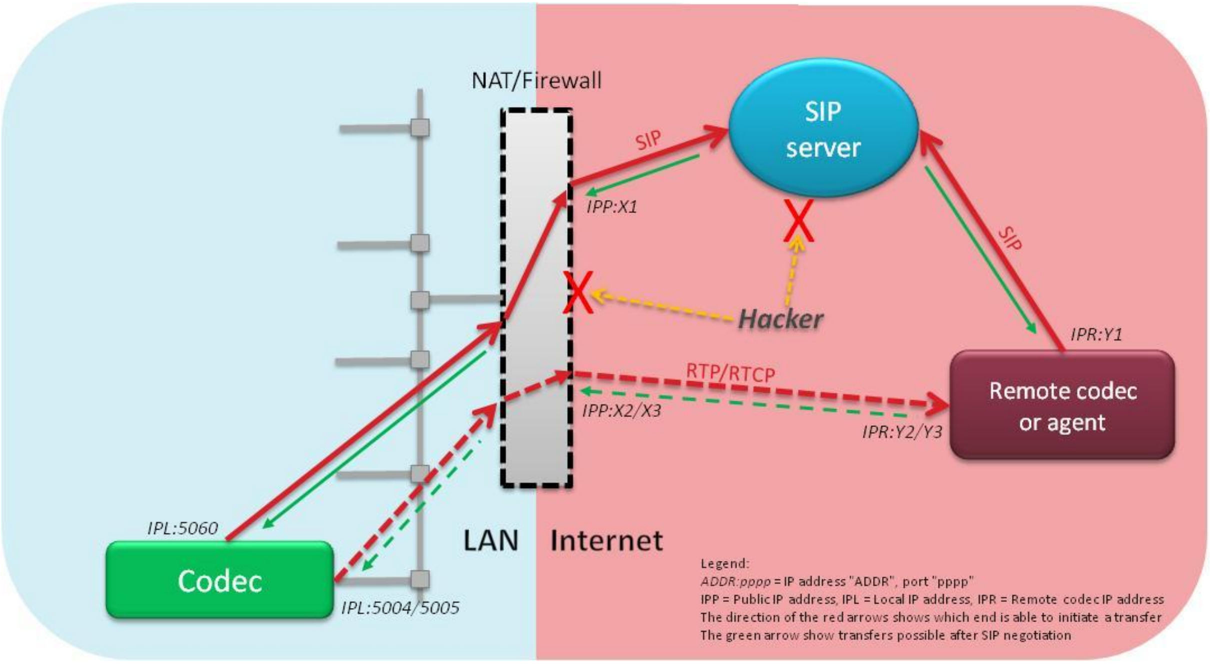
Various alternatives or additions can improve the operation and/or security:

- In order to use several codecs on one site, allocate to each one a unique port number for SIP, instead of the standard 5060. *If needed, the SIP port number of an AETA codec is editable as well.*
  In this way, only one public IP address is "consumed", and a remote agent can select the destination by specifying the port number instead of using the implicit 5060.
- Even with only one codec, it is interesting to set up a port other than 5060 or 5061. This will make it slightly more difficult to call from remote, but this reduces the vulnerability to undesired calls. Hackers use to scan by priority the standard ports 5060 and 5061. However this delays but does not block attacks!
- If attacks are experienced, it is possible to add rules in the firewall and blacklist their source IP addresses. That is a temporary protection, however.
- A white list is more efficient: if it is possible to define a restricted list of remote IP addresses entitled to set up links, configure the firewall for accepting data only from these addresses (to the forwarded ports).
  However, beware not to block the STUN requests!

5.6. *NAT and SIP server*

5.6.1. Principle

To overcome the main drawbacks of the methods described previously, a good solution is to implement infrastructure entities such as a SIP server. Of course, that is only valid for codecs using the SIP protocol. In the following case the server is located on the Internet side:


NAT/Firewall
SIP
SIP server
IPP:X1
SIP
Hacker
IPR:Y1
RTP/RTCP
IPP:X2/X3
IPR:Y2/Y3
Remote codec or agent
IPL:5060
Codec
IPL:5004/5005
LAN
Internet
Legend:
ADDR:pppp = IP address "ADDR", port "pppp"
IPP = Public IP address, IPL = Local IP address, IPR = Remote codec IP address
The direction of the red arrows shows which end is able to initiate a transfer
The green arrow show transfers possible after SIP negotiation

The packets which should go through the access router divide in two types:

- Signaling messages, using the SIP/UDP protocol.
- Compressed audio streams with RTP/UDP protocol.

With this architecture, the codec on the LAN is always the device that opens a path through the router:

- SIP requests for registering the codec on the SIP server; afterwards the codec/server communication path is maintained by using a periodic “keep alive” process.
- STUN requests[1] for the codec to discover the address and ports for the RTP transfers.

At registration time, the server checks the codec identity (authentication with SIP account and password) and keeps track of the presence and location (public IP address and SIP port) of the agent. A transaction initiated by any agent systematically goes through the SIP server.

Conversely, the RTP streams can be exchanged directly between agents without necessarily transiting through the server.

With this architecture there is no need for any fixed inbound route (port opening and forwarding), because the server dynamically deals with the public port allocation.

[1] Not shown here to simplify the diagram

5.6.2. Advantages

- Once the SIP server is installed, installing a codec on a site is rather simple.
- The codec is identified and reachable by its SIP URI, fixed because bound to its SIP account. Therefore changing IP addresses is not an issue, including when the geographic location of a codec is changing.
- There is nothing against setting several codecs on the same site.
- No need to set the NAT router in a specific way (forwarding, etc.).
- Agents which call or are called must be authenticated beforehand to be registered. Undesirable agents cannot intrude by initiating a session without being authenticated[1].
- The codec can also communicate directly with other units in the same LAN.
- There is no need to create breaches by statically opening ports that intruders might take advantage of. The outside cannot simply initiate a session, and even less reach the control interface of a codec, unless this was enabled on purpose, for remote management needs.
- Even a mobile network access can be used to link a codec, although the frequent restricted NAT issue should be dealt with appropriately (see further).

5.6.3. Drawbacks

- For a start a SIP server must be available with SIP accounts and URI; installing one is rather complex. In addition the server is a unique point of failure for the whole system, thus it must be secured appropriately.

⇨ *However one can lean on a third party for this; for example AETA offers a SIP server that is reliable and dedicated to broadcast applications.*

- There exists some cases where the network operator blocks SIP, either at the network level, or in the router; some examples met sometimes:
  + Port 5060 blocked in the access router (provided and managed by the operator)
  + RTP streams blocked by a mobile network (VoIP blocked by operator)
  In such case one should, if possible, ask the operator to unblock the service.
- In a similar way, the operation may be hampered or blocked by a too restrictive firewall.
- There is a possibility for an external entity to detect the public ports (dynamically allocated) during legitimate transactions and try to intrude via these routes.

[1] To be totally accurate, in fact that *is* possible, if the server is set to allow such "permissive" behavior. But that is not common policy, and obviously that is not recommended for a safe operation.

5.6.4. Recommendations and alternatives

Although it implies a significant effort for the initial phase, such architecture brings big advantages, on the functional ground first, but also for security, which is our concern in this document. A very good security level can be reached for a moderate global cost, especially if adding some adjustments:

- Do not add port forwarding (like with the method described in 5.5), which is useless in this architecture and can only create vulnerability.
- Conversely, make sure the firewall is not too restrictive and does not prevent the routes to the server to be opened, as well as those to the STUN server, and the RTP streams to the remote codec during a session.
- It can be interesting to set the SIP server with a non standard SIP port (other than 5060 or 5061). Very often this allows to work around blocking by the operator equipment.

⇨ *With the AETA SIP server you can use such an alternate port.*

- To increase security it is possible in the firewall to allow only the SIP server's IP address for any transfer with the SIP protocol/port (and the STUN server's IP address for STUN requests).
  In such way any attempt to set up an undesired session is blocked in any case.

5.6.5. Restricted or symmetric NAT case

In the special case of a router with "restricted" or "symmetric" NAT, the ports found using STUN are not valid for the subsequent transfers with the remote codec.

Some SIP servers can work around this issue by coupling with an RTP proxy. In such case, the server first detects the restricted NAT situation at the codec level. Then during the session, instead of exchanging the streams in a peer to peer scheme as usual, the RTP proxy will relay the packets and ensure the end to end routing of the packets. This is done at the cost of a "detour" of the streams via the proxy, but the link is fulfilled in spite of the restrictive NAT(s) on the way.

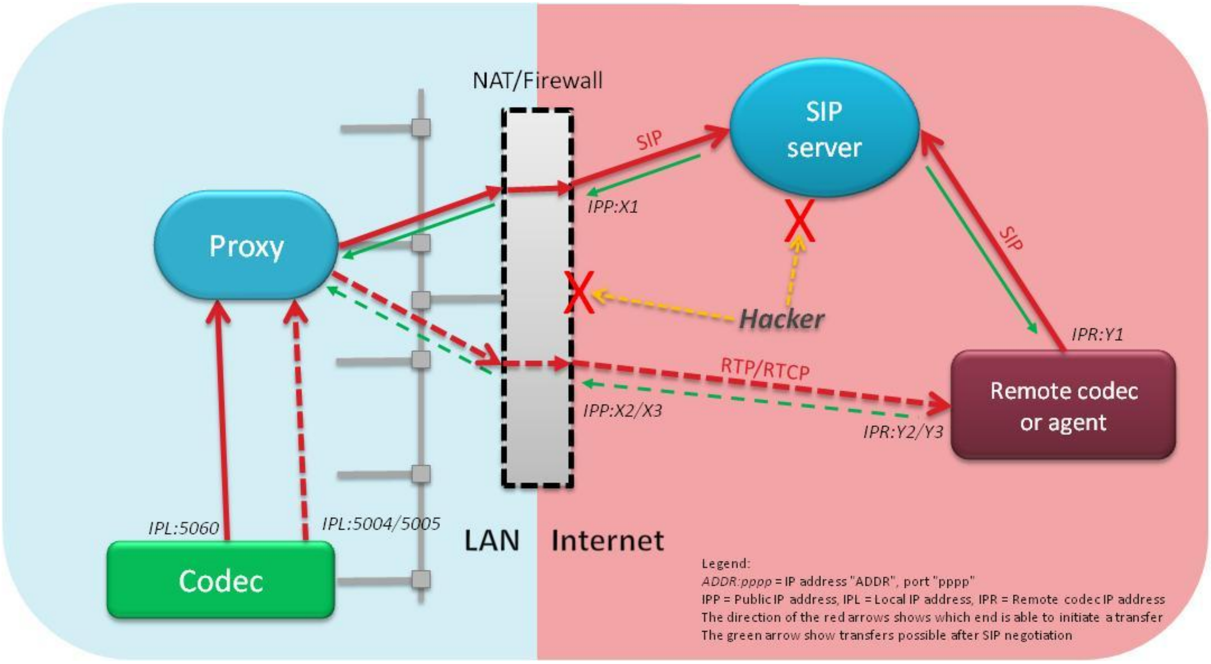
This solution is especially useful for these situations where restricted NAT cannot be avoided, as almost always on a mobile network access.

⇨ *The AETA SIP server also provides this feature.*
More generally, the AETA server is a good way to set up quickly an organization based on using a SIP server, without having to bother with a complex installation, and without investing in the required hardware resources and the related maintenance.

5.7. *NAT, SIP server and proxy*

5.7.1. Principle

Starting with the previous architecture, the security can be improved further by adding on the local site a *proxy* that comes as an intermediary for session with outside units.



Legend:
ADDR:pppp = IP address "ADDR", port "pppp"
IPP = Public IP address, IPL = Local IP address, IPR = Remote codec IP address
The direction of the red arrows shows which end is able to initiate a transfer
The green arrow show transfers possible after SIP negotiation

All the signaling is relayed via this proxy, but it does not manage presence and registration, which stay under control of the SIP server. It also relays RTP streams. Only the proxy makes transfers directly with the outside.

- The firewall may block completely the *direct* outbound access to all the codecs.
- Conversely, it can specifically entitle the proxy to pass through the router, only for those useful protocols and reserved ports.

5.7.2. Advantages

In addition to the advantages related to using a SIP server:

- Very high security: the codecs are totally isolated from the Internet. Even the proxy is exposed only just as needed for the essential routes.

5.7.3. Drawbacks

- This makes one more device to manage. Though its configuration is not necessarily very complex (no user account data base to maintain).
- Configuring the codes is slightly more complex.

⇨ *Namely, on AETA codecs, the IP address of the proxy must be entered among the SIP parameters. Caution, remind that, for a "mobile" codec, this setting is location-dependent!*

5.7.4. Recommendations and alternatives

The same recommendations apply here as for the previous case.

When the SIP server is located inside the organization's network, it makes sense to combine in the same physical unit the SIP server and the proxy. Many SIP servers readily include such capability.

5.8. NAT, SIP server and SBC

5.8.1. Principle

An SBC (Session Border Controller) is a device that groups in a consistent entity the Internet access, firewall and SIP proxy functions. It is then set up in place of the router/firewall of the architectures described so far.

The SBC applies extended control over the VoIP/AoIP sessions across the Internet access, with a large variety in the amount of implication and the supported functions. It allows to achieve a maximum security level, and this type of tool is especially used for securing organizations with a large VoIP management capacity.

For a more detailed description of the functions and techniques implemented in an SBC, one can refer to the following page:

http://en.wikipedia.org/wiki/Session_Border_Controller

5.8.2. Advantages

- Maximum security against all types of attack.
- Numerous capabilities for fine adjustment to the application needs and environment (selection of entitled external resources, filtering of allowed session types, etc.).

5.8.3. Drawbacks

- The necessary equipment is expensive: for this reason this solution is not suitable for small size organizations.
- Installing and managing an SBC is rather complex, and require a skilled network manager. Like for a sophisticated firewall, having an insufficient understanding of the system can lead to inadvertent security breaches, or conversely to excessive blocking that hampers the operation.

AETA can give advice if you wish to study such a system (please consult our sales team).

6. Conclusions

This overview is partial and simplified on some areas, for the sake of clarity. But its goal is to bring out the principles usually applied and the main issues raised in the installations.

Among the main outlines, there is a trivial principle: nothing is free. The solutions that look very easy to set up are either unreliable or unsafe. Eventually, after adding workarounds one reaches a system which is not really simple any more.

Conversely, provided an initial effort is done to base on a reliable infrastructure, the daily operation can be really simple, and without compromising the security.