SoliCall PBXMate

User Manual

1 <u>INTRODUCTION</u>

Does your VoIP network suffer from poor audio quality? Unfortunately, in most cases the answer is YES. Different reasons cause this problem and some of them are due to external factors that you cannot control such as echo and noise originated from the far end. SoliCall's **PBXMate** will allow you to gain control over the audio quality in your network.

What is the PBXMate?

The PBXMate is a software product that improves audio quality for all participants in a call. The PBXMate constantly improves and monitors quality.
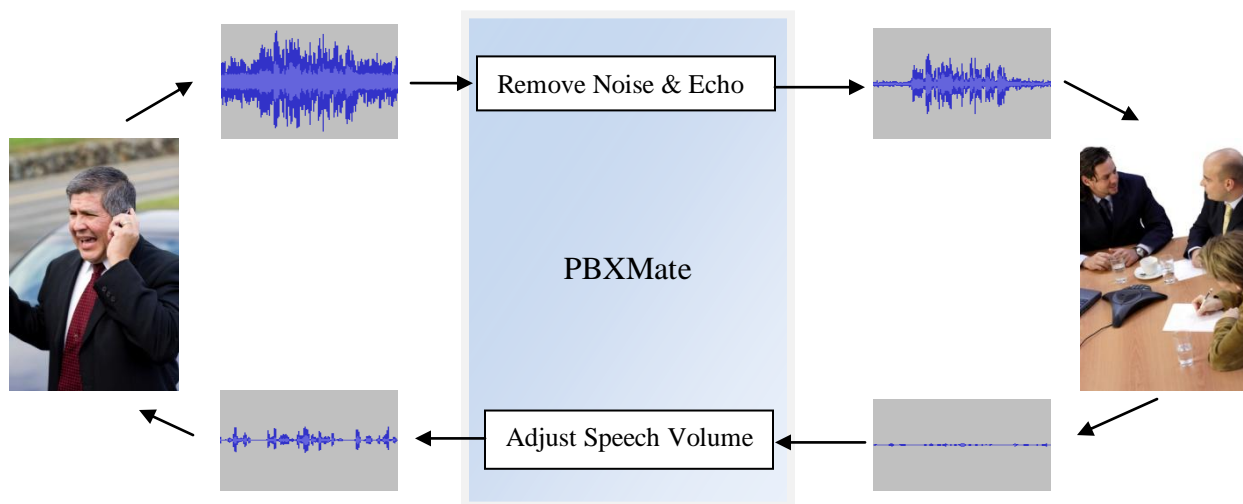
Noise and echo are removed and speech level is adjusted to a comfortable level

General			Quality (Extreme)												
			Origin -> Destination						Destination -> Origin						
Origin	Destination	Duration	Noise Level	Max AGC Coef	Echo Level	Jitter (ms)	Packet Loss (%)	Min Delay (ms)	MOS Score	Noise Level	Max AGC Coef	Echo Level	Jitter (ms)	Packet Loss (%)	Min Delay (ms)	MOS Score
3@192.168.0.113	2@192.168.0.145	36	16	100	720	0	0.11	44	4.31	9728	500	1	0	0.15	31	4.29

The PBXMate displays real-time statistics on the quality of the calls

Use Cases

- Remove echo your users hear when making long distance calls or calls over WiFi.
- Shield your customers from the ambient noise in your call center.
- Enhance the audio quality in conference calls with multiple participants.
- Reduce background noise coming from external cellular phones.
- Maintain a comfortable audio volume at conference rooms regardless of the distance between the speakers and the microphone.
- Improve the accuracy of speech recognition engines.
- Monitor calls quality to alert, in real time, on low quality calls.
- Record calls going through the network.

Plug-And-Play Architectures

The PBXMate supports three plug-and-play architectures.

PBXMate as a SIP Trunk (the default architecture)
PBXMate registers as SIP Trunk to the PBX. When the PBX receives a call that needs to be filtered, it routes it to the PBXMate SIP Trunk which, in return, dials back to the PBX. Using the dial-plan in the PBX, the administrator controls which calls are filtered.
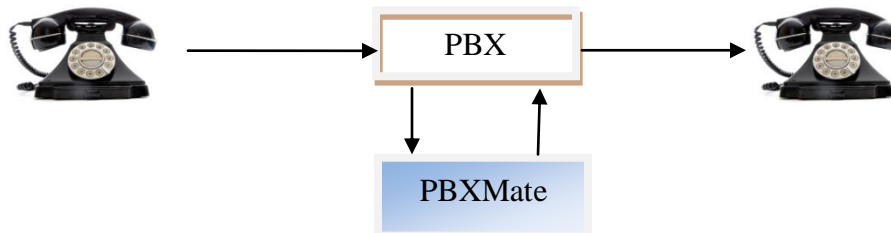
PBXMate as a SIP Proxy

The PBXMate acts as the Sip Proxy for the internal IP Phones. It filters all the calls of these phones. This architecture is **fully transparent** to the PBX and there is no need to do any change in the PBX's dial-plan or in any other network component.
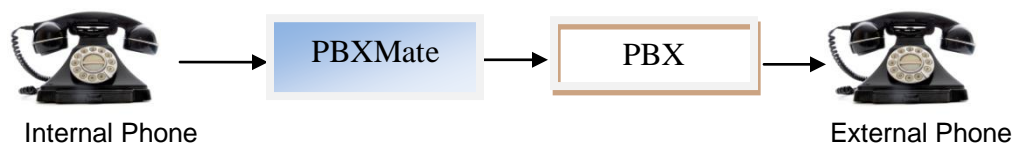
<u>PBXMate as a Router</u>

The PBXMate routes the calls to their destination based on a simple routing plan. This architecture is very similar to the first one (Sip Trunk), but in this case the PBXMate dials to the destination instead of dialing back to the PBX. This architecture can be used when due to licensing issues you do not want to increase the total number of calls in your PBX.

Scalability

The PBXMate can handle hundreds of concurrent calls on a single server. A single PBXMate process splits the load between multiple threads. In addition, multiple instances of the PBXMate can be run on the same server and on multiple servers. The exact number of concurrent calls that can be supported on a single server depends on: the desired 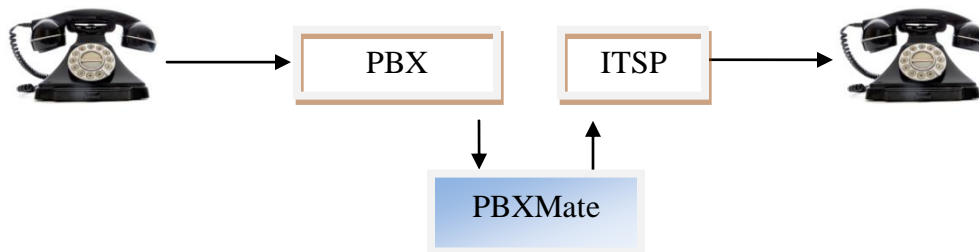configuration (e.g. single side filtering vs. double side filtering), the algorithms that are enabled (e.g. noise reduction, AEC, basic AEC, AGC) and the hardware specifications.

24/7 Uptime

The PBXMate is a robust product which contains a built in recovery & instance mechanism to support 24/7 up time. In addition, in case the PBXMate is being shut down, the PBX will automatically divert all incoming calls directly to their destination without any downtime to the VoIP network.

Supported Networks

As the PBXMate uses the SIP and RTP standards, it works with all VoIP networks that can support these standards. There is also an option to integrate the PBXMate with H.323 network – please consult the user manual for more details.

Supported Platforms

The PBXMate can run on both Linux and Windows. It has both 32bit version and 64bit version. It can be configured to run as a service (Windows) or daemon (Linux). The PBXMate has a flexible port mapping that allows it to run on the same machine that is running the PBX.

Additional Technical Facts

- Supports both web based graphical interface and command line interface.
- Equipped with algorithms to overcome packet loss & jitter.
- Full support for video in two modes: bypass or multiplexed on existing ports.
- Can record all statistics to a file for offline analysis.
- Can fully impersonate the originating SIP phone.
- Adds a minimal delay, about 16ms, to the call.
- Filtering mode can be changed during the call via DTMF controls.
- Allows registering voice sample for additional enhancement in audio quality.
- Can be activated in monitoring mode – i.e. no filtering will be made to the audio.
- Supports different configuration per caller id.
- Allows denial of service for specific caller ids. The PBXMate will return a 503 SIP message in these cases.
- The basic version supports G.711 & G.722. For additional codecs, please contact support.
- Can be configured as a hosted solution.
- Includes a detailed user manual and handful configuration samples & tips.

Customer's Experience

The PBXMate is successfully improving audio quality in many production environments. It is being used by many companies that were unwilling to compromise on quality, ready to invest in order to increase customer satisfaction and therefore add to their competitive edge. Among our customers are call centers, mobile operators and unified-communication platform providers. Case studies and reference accounts are available upon request.

2 <u>QUICK START</u>

2.1 Unpacking & Environment Configuration

- Select a computer in your network and extract the package to any directory you choose. For the rest of this manual we assume that on Linux you extracted the package in the directory "/usr/local/solicall" and on Windows you extracted the package in the directory "c:\solicall"
- Define **SOLICALL_BIN** environment variable to point to the bin sub-directory. The environment variable should end with "/" for Linux or "\" for Windows. Make sure the environment variable is defined automatically for every new shell.

  On Linux this is done in the ".bashrc" file or equivalent:

  ```
  export SOLICALL_BIN=/usr/local/solicall/bin/
  ```

  On Windows this is done via the control panel->System->Advanced (tab) ->Environment Variables (button):
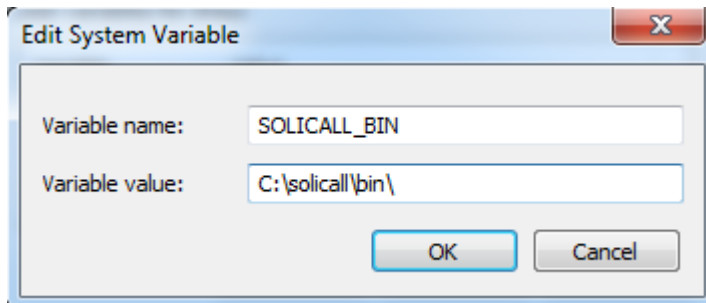
- **On Linux**, based on your architecture/distribution, copy the four executable files from the appropriate location under *arch* sub-directory to the *bin* sub-directory. Make sure all four files (*solicallpbxmate*, *solicall_guard*, *solicall_cli* & *solicall_gui)* have executable permission (*chmod +x <file name>*). If your architecture is CentOS 5 or RHEL 5 then use the fedora-6 binaries. If you architecture is CentOS 6 or RHEL 6 then use the fedora-12 binaries.

2.2 Graphical Interface Vs. Command Line Interface

- The PBXMate is delivered with both graphical interface (solicall_gui) and command line interface (solicall_cli). For details on using the command line please see chapter 3.3.
- The graphical interface give you access only to the basic parameters & features. For advanced options you should use the command line interface and edit the configuration file "pbxmate.conf".
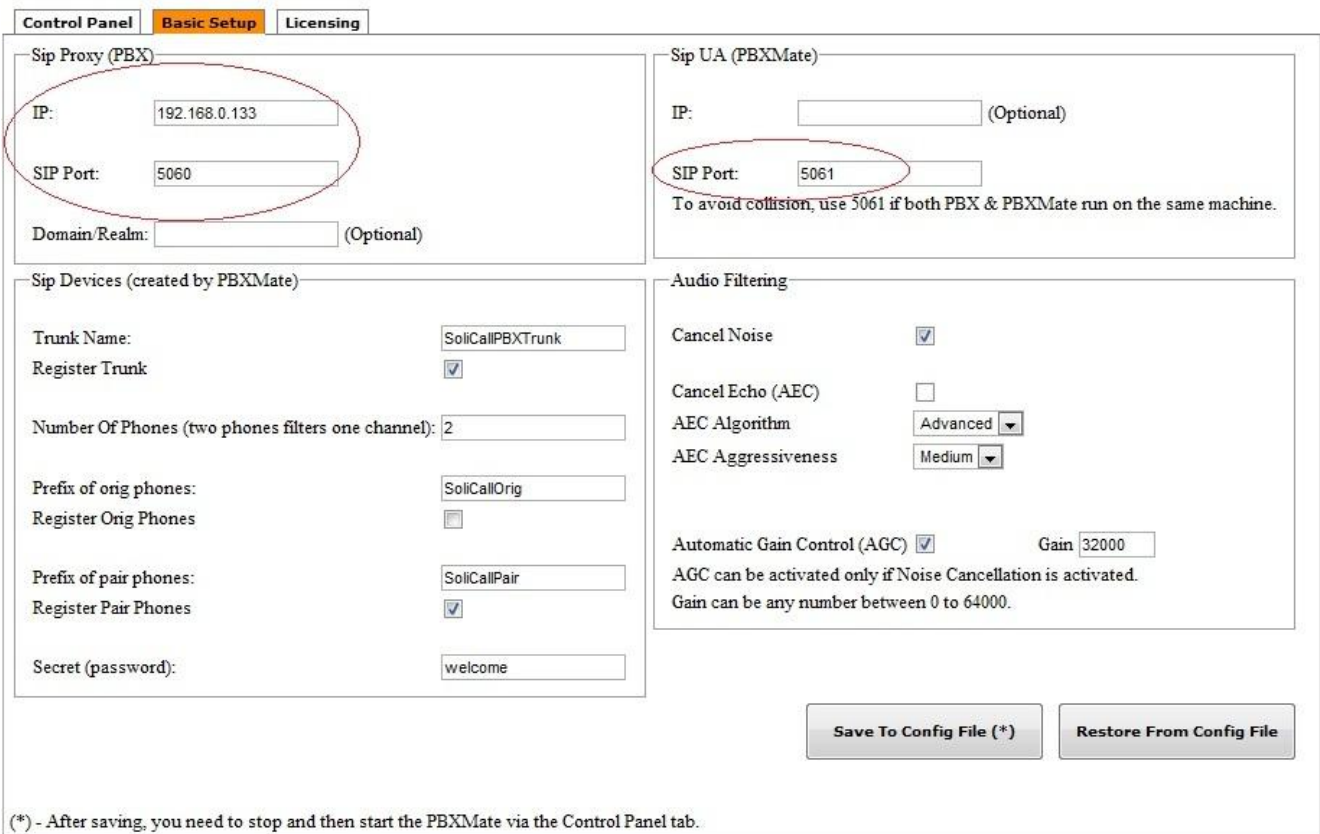
2.3 PBXMate Configuration

- Open a cmd/terminal window. On Linux *open a terminal window*. On Windows this is done via *Start->Run->Cmd*.

- Navigate to the bin directory and execute

    - On Linux: *./solicall_gui -root /usr/local/solicall -ports 8083*.
    - On Windows: *solicall_gui -root "c:\solicall" -ports 8083*.

    You should get back the following message:
    Mongoose 2.8 started on port(s) [8083], serving directory…

- Open your browser at http://127.0.0.1:8083/bin/solicall_gui.html.
- Note: It is recommended to use screen resolution 1024x768 or higher.
- Go to the Basic Setup tab and update the parameters. You need to pay attention to the below marked parameters.
- By default AEC is disabled. You need to enable it if you require echo cancellation.

- If you are not using the graphical interface and also in order to configure more advanced parameters you can manually edit the configuration file "pbxmate.conf".

2.4 PBXMate Activation

- In your browser go to the Control Panel tab and press the **Start** button.

- Press the "Manual Refresh" button and monitor the changes in the Status line, System Log & Application log.

- Alternatively, If you do not to use the graphical interface, run solicall_cli and when you get the prompt write "**start**" to activate the PBXMate. When you get the prompt back write "**status**" to see the status of the application.

2.5 Monitoring the Logs

- In the application log you should see a line like:

02/14/12 10:35:36 INFO in pbxmate_main, **Loading PBXMate** version 1.6.40

- If you do not see such a line please, check for errors in the system log or application log.

- Alternatively, if you do not to use the graphical interface, you can manually look at the log files. The application log is the file "errLog" under the log directory. The system log is the file /tmp/SoliCallTmpLog (on Linux) OR c:\SoliCallTmp\SoliCallTmpLog (on Windows).

If you plan to use PBXMate in a SIP Proxy architecture, then please skip to chapter 4.7

2.6 Complete Registration of the PBXMate with your PBX

- By default the PBXMate tries to register the following component with your SIP PBX:
    - o SoliCallPBXTrunk - a SIP trunk to receive calls from your PBX.
    - o SoliCallPair0 – a SIP phone to initiates calls.

- You need to configure your PBX to allow the above components to register. Please see chapter 4 for examples.

- In the application log you should see a line like:

    02/14/12 10:35:38 INFO in MySip::mainLoop, **All phones are active**

- If you do not get this message, it means that the PBXMate was not able to complete the registration. In such case please review the following:
    - o If passwords are required by your PBX, is the *Secret* (password) parameter configured correct?
    - o If your PBX requires SIP phones to register by numbers (and not by names), change *PairPhonesNamePrefix* parameter to be a number. For example if you set *PairPhonesNamePrefix=50* then the SIP phone will register as 50**0** (the last zero is automatically added at the end to indicate that this is the first phone).

2.7 Diverting Calls and Monitoring Statistics

- In your PBX divert the calls, you want to filter, to the SoliCall trunk. Please see chapter 5 for examples.
    - o In Asterisk it can be done, for example, via the following commands:

    exten => _X!,1,Dial(SIP/SoliCallPBXTrunk/${EXTEN})

- Whenever the PBXMate receives a call:
    - o If the max number of concurrent calls was not reached, it will automatically initiate another call to the original destination of the call.
    - o Otherwise, if the max number of concurrent calls was reached, it will reply with the SIP error "486 (Busy Here)".

- For every call that is filtered by the PBXMate, statistics information is being written to the file "stats" (under the log directory).

- In your browser navigate to http://127.0.0.1:8083/log/RealTimeAverageStats.html and http://127.0.0.1:8083/log/RealTimeExtremeStats.html to see graphical presentation (average & extreme) of the current active calls:

Real Time Extreme Statistics On Active Calls

Total Number Of Active Calls: 1

General			Quality (Extreme)												
			Origin -> Destination							Destination -> Origin					
Origin	Destination	Duration	Noise Level	Max AGC Coef	Echo Level	Jitter (ms)	Packet Loss (%)	Min Delay (ms)	MOS Score	Noise Level	Max AGC Coef	Echo Level	Jitter (ms)	Packet Loss (%)	Min Delay (ms)	MOS Score
3@192.168.0.113	2@192.168.0.145	36	16	100	720	0	0.11	44	4.31	9728	500	1	0	0.15	31	4.29

2.8 Testing the audio filtering

After you have configured the system correctly, you can test the audio filtering (noise reduction, echo cancellation & gain control). You can dial, for example, to an external mobile number that is located in a noisy environment. Then using DTMF you can toggle deactivate/activate of the filtering:
- *9 to deactivate filtering.
- #9# to activate filtering.
After each operation you should hear a short confirmation tone.

The toggle will not perform correctly if the call is between two phones that are both connected to the PBXMate as their SIP Proxy.

The configuration file, pbxmate.conf, includes many parameters for tuning the filtering. SoliCall support team will be happy to help you and recommend on the best tuning to your system. For more information see chapter 6.7.

3 <u>CONFIGURATION DETAILS</u>

3.1 Directory Structure

Create a directory called SoliCall. It should contain the following subdirectories:
- bin – in this directory should include:
  - solicallpbxmate - the main executable.
  - solicall_guard – the guard for the main executable.
  - solicall_cli – CLI, command line interface.
  - solicall_gui – GUI, graphic user interface.
  - pbxmate.conf – the configuration file.
  - the license file – required for non-evaluation version.
  - Misc files like special tones.
- log – in this directory error log and statistics will be written. The application will automatically write all errors to an errLog file in this directory.
- rec – to store recording data.
- registration – to store registration data.
- arch – executables per architecture/distribution (Linux only).

3.2 Environment Variable

You need to define the environment variable SOLICALL_BIN and have it pointing
to the bin directory (e.g. /usr/SoliCall/bin/). It is also recommended to add this directory to the path.
The environment variable should end with "/" for Linux or "\" for Windows

3.3 Using the command line

After running solicall_cli you can give it the following commands:
- status – to query the status of the PBXMate.
- start – to activate the PBXMate.
- stop - to stop the PBXMate.
- shutdown – to make PBXMate stop accepting new calls. Available in Linux version only.
- show_bindings – to see details on all phones that registered with us.
- exit – to exit the CLI.

4 **EXAMPLES**

4.1 Configuring Asterisk using Sip Trunk

For the purpose of this example we assume you have a simple dialplan for a context named "internal". We assume your simple dialplan looks like this:

```
[internal]
exten => _X!,1, Dial(SIP/${EXTEN})
```

In *sip.conf* you need to define the SoliCall trunk & the SoliCall pair phone:

```
[SoliCallPBXTrunk]
type=friend
context=internal
host=dynamic
dtmfmode=rfc2833

[SoliCallPair0]
type=friend
context=SoliCall
host=dynamic
dtmfmode=rfc2833
```

In *extensions.conf*, you need to change your simple dialplan as follows:

```
[internal]
exten => _X!,1,Dial(SIP/SoliCallPBXTrunk/${EXTEN})
exten => _X!,2, Dial(SIP/${EXTEN})
```

This change will divert all calls to the SoliCall trunk. In case the SoliCall trunk is not responding or reached its limit your regular dialplan will be executed.

In *extensiona.conf*, add the SoliCall context:
```
[SoliCall]
exten => _X!,1, Dial(SIP/${EXTEN})
```

This will cause all calls coming back from the SoliCall trunk to continue executing your regular dialplan.

4.2 Configuring Asterisk using Sip Trunk and Impersonation

In some cased, you would like the PBXMate to fully impersonate the caller id of the originator of the call. This is important if you need PBXMate to be fully transparent.

For the purpose of this example we assume you have a simple dialplan for a context named "internal". We assume your simple dialplan looks like this:

```
[internal]
exten => _X!,1, Dial(SIP/${EXTEN})
```

Please edit *pbxmate.conf* file and make sure you have the following settings:

```
RegisterTrunk=1
RegisterPairPhones=0
PairPhoneShouldImpersonateTheOriginator=1
```

In *sip.conf* you need to define SoliCall trunk:

```
[SoliCallPBXTrunk]
type=friend
context=internal
host=dynamic
dtmfmode=rfc2833
```

In *extensions.conf*, you need to change your simple dialplan as follows:

```
[internal]
exten => _X!,1,GotoIf($["${CHANNEL(useragent):0:8}" = "SoliCall"]?3:2)
exten => _X!,2,Dial(SIP/SoliCallPBXTrunk/${EXTEN})
exten => _X!,3,Dial(SIP/${EXTEN})
```

After this change, all calls will be routed to SoliCall trunk unless their user-agent is SoliCall in which case the calls will be routed to their original destination.

Please note that old versions of Asterisk (up to 1.4) do not support the variable **CHANNEL(useragent)**. For these old versions you can use instead the variable **SIPUSERAGENT** or **SIPCHANINFO(useragent)**.

After the above changes, the PBXMate will fully impersonate the phone that originated the call. Therefore, Asterisk will think that this phone is now initiating two calls. If in your configuration Asterisk has a limit of one call for the originator phone, you will need to increase the limit to two calls. This can be simply done in *sip.conf*:

```
call-limit=2
```

If when making a call you get an authentication error, it means that Asterisk is trying to authenticate the call coming from PBXMate against the data of the originator phone. There are two solutions to this problem:

In *sip.conf* define that the originator phone does not have to authenticate the INVITE message:

```
insecure=invite
```

Alternatively, you can change the originator phone to type=peer. After doing this change, Asterisk will do not match the From field in the INVITE message but will match the IP. As a result the authentication will be done properly against the data of the PBXMate.

4.3 Using channel variables when routing calls in Asterisk

If your dial-plan is using channel variables (e.g. DNIS & CHANNEL) to route the call, their value will be lost when the PBXMate is making the second call. Therefore, you need to send this information when dialing to the PBXMate so it can be extract when receiving the second call.

For example, if your original *extensions.conf* looks like this:
exten => _X!,1,Dial(SIP/${DNIS}${CHANNEL})

We will use two new variables STORE_DNIS & STORE_CHANNEL to store the correct values of these two parameters.

The modified dial-plan looks like this:
exten => _X!,1,Set(STORED_DNIS=${DNIS})
exten => _X!,2,Set(STORED_CHANNEL=${CHANNEL})
exten => _X!,3,GotoIf($["${CHANNEL(useragent):0:8}" = "SoliCall"]?7:4)
; Dial to PBXMate with the accumulated parameters
exten => _X!,4,Set(ACCUMULATIVE_EXTEN=${DNIS}${CHANNEL})
exten => _X!,5,Dial(SIP/SoliCallPBXTrunk/${ACCUMULATIVE_EXTEN})
; The Dial to PBXMate failed, so dial to the final destination
exten => _X!,6,Goto(9)
; We got a call from PBXMate, so extract the parameters (assuming DNIS is a four digit number)
exten => _X!,7,Set(STORED_DNIS=${EXTEN:0:4})
exten => _X!,8,Set(STORED_CHANNEL=${EXTEN:4})
; Dial to the final destination
exten => _X!,9,Dial(SIP/${STORED_DNIS}${STORED_CHANNEL})

Please note that it is usually recommended not to have special characters like "/" and ":" in the ACCUMULATIVE_EXTEN variable.

4.4 Configuring Elastix

Since Elastix PBX engine is similar to Asterisk, you need to follow the examples for Asterisk with the following changes:
- Instead of changing the file *sip.conf*, change the file *sip_custom.conf*
- Instead of changing the file *extensions.conf*, change the file *extensions_custom.conf*

4.5 Configuring FreeSWITCH using Sip Trunk

A detailed case study is described in
http://wiki.freeswitch.org/wiki/PBXMate-FreeSWITCH-integration

4.6 Configuring 3CX using PBXMate as a Router

A detailed case study is described in
http://www.solicall.com/wp/3CX-SolicallPBXMate_Solicall.pdf

4.7 Using PBXMate as a SIP Proxy without changing the PBX

This example illustrates activation of the PBXMate as a SIP Proxy for the IP Phones. In this architecture **there is no change in the PBX itself** and it is **fully transparent** to the PBX. In this architecture, the PBXMate could be viewed as an add-on to the IP-Phones. In this architecture every call to/from the IP Phones that are registered to the PBXMate ALWAYS pass via the PBXMate. An internal call between two phones that are registered to the PBXMate pass via the PBXMate twice and therefore counted as two calls in the PBXMate.

You need to do the following changes in pbxmate.conf:

```
RegisterTrunk=0                              ← Since we are not using trunk in this example
RegisterPairPhones=0                         ← Since the PBXMate is fully transparent
NumPhonesToAllocate=6                        ← Assuming you want to support up to 3 concurrent calls.
PairPhoneShouldImpersonateTheOriginator=1
InviteShouldMatchPairPhoneName=0
SipProxyArchitecture=1
```

In addition, you need to change in IP Phone the address of their SIP Proxy to be the address of the PBXMate (instead of the address of the PBX). You can verify that the IP Phones have registered successfully by typing the command "show_bindings" in the CLI of the PBXMate.

In some rare scenario this method might not work. This can happen if you are using passwords and the URI contains the IP of the phones and your PBX is trying to match the URI in the encrypted response. In such rare cases, you will have to provide the passwords to the PBXMate – see 4.8.

4.8 Using PBXMate as a SIP Proxy with password information

You should use this example in the rare scenario in which the previous example (4.7) does not work for you. Let's assume you have two phones 201 & 202 and phone number 202 can handle two calls simultaneously. In addition let's assume that their password is w201 & w202.

You need to do the following changes in pbxmate.conf:

```
RegisterTrunk=0                ← Since we are not using trunk in this example
RegisterPairPhones=1
NumPhonesToAllocate=6          ← To support the following three phones/lines.
PairPhonesName=201;202;202;    ← Note that 202 appears twice as it has two lines
PairPhonesSecret=w201;w202;w202;
```

OrigPhonesName=201;202;202;   ← The same as PairPhonesName
OrigPhonesSecret=w201;w202;w202;   ← The same as PairPhonesSecret
PairPhoneShouldImpersonateTheOriginator=1
InviteShouldMatchPairPhoneName=1
SipProxyArchitecture=0

In addition, you need to change in IP Phone 201 & 202 the address of their SIP Proxy to be the address of the PBXMate (instead of the address of the PBX). You can verify that the IP Phones have registered successfully by typing the command "show_bindings" in the CLI of the PBXMate.

In order to verify that the PBXMate has successfully registered to the PBX on behalf of the phones, in the application log you should see a line like:

10/15/09 19:36:44 INFO in MySip::mainLoop, **All phones are active**

5 UNDERLINE THE HOOD

Background

SoliCall PBXMate was designed to work with any existing VoIP Network. Therefore, the PBXMate is a stand alone application that works with the telephony network using a **standard interface**. PBXMate can be run on any Linux/Windows machine on the network.

The current audio path in the network looks like this:
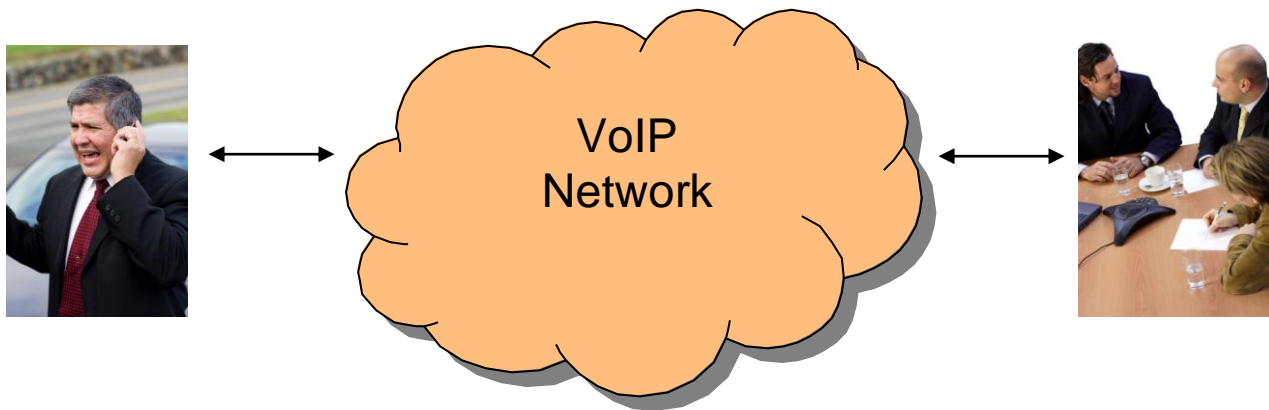
When filtering the call via PBXMate the audio path looks like this:
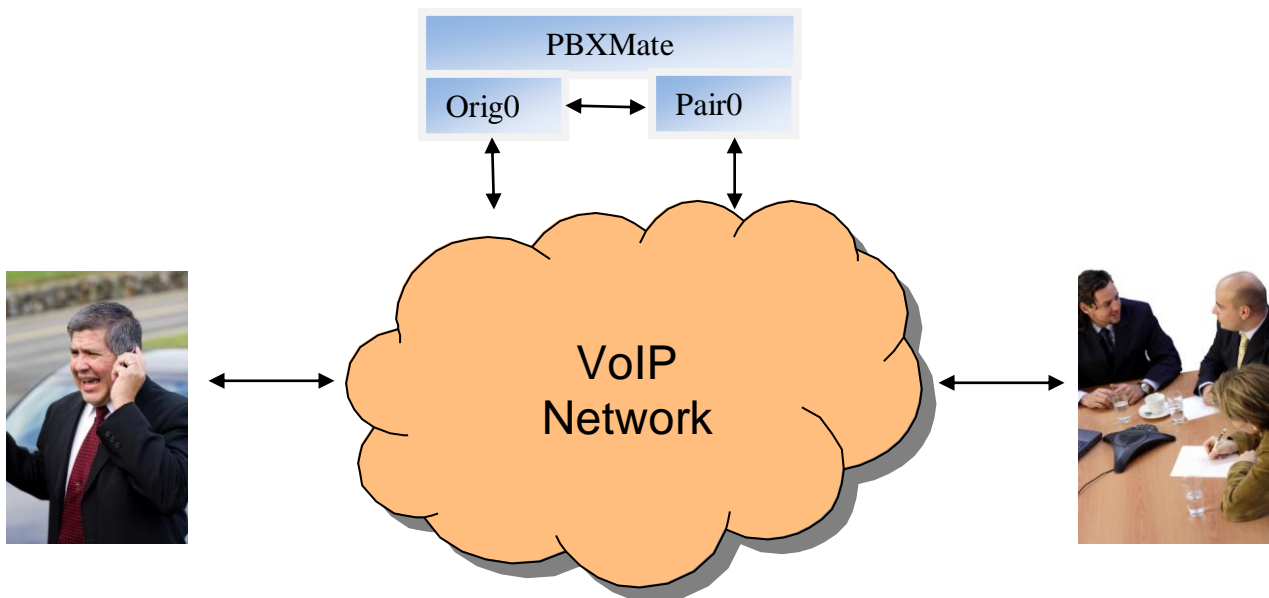
"Orig0" and "Pair0" are acting as two SIP Phones.

Call Setup Example

When a new call is entering to the network and it is decides that it should be filtered (e.g. it is coming from cellular phone), the network (e.g. using dial plan) diverts the call to SoliCall Trunk (or specifically to the phone "Orig0"). When SoliCall receives the call, "Pair0" will dial to the original destination therefore completing the desired audio path.

Trunk & Phone Pairs

The PBX Mate works with a single SIP trunk (SoliCallPBXTrunk) and multiple phone pairs:
- SoliCallOrig0 & SoliCallPair0.
- SoliCallOrig1 & SoliCallPair1.
- Etc.

The exact SIP name of the trunk & phones is configurable.

Filtering Type

When a call enters the PBXMate, one of the following filtering types could be assigned to it:
- 0 – No filtering is required.
- 1 – Filter only the audio coming from the originator of the call.
- 2 – Filter only the audio coming from the destination of the call.
- 3 – Filter audio coming from both sides of the call.

Note: that the filter is either to **remove noise** and/or to **remove echo** (depending on the parameters "NeedToDoNoiseCancellation" & "NeedToDoAEC" in the configuration file).
If you remove noise is enabled, then AGC can also be activated.

Command

For each call that enters the PBXMate, it should receive a command that tells it:
- The destination of the call. i.e. where the PBXMate should dial.
- Filtering type to use for the call.

A command might look like:
- *--3
  This command means to dial to the original destination of the call and to use filtering type of 3.
- 104@internal—1
  This command means to dial to 104@internal and to use filtering type of 1.

Whenever a call arrives to SoliCall, it looks for a suitable command in the following order:

- It looks if a command was added to the caller id. I.e. in this case the command is part of the caller id of this call. This option is rarely used.

- It tries to match the caller id to predefined expressions that are written in the configuration file. For example if the configuration file contains the following:

    CmdBasedOnCallerID=1??@192.168.1.241=*--3

    Whenever a call enters SoliCall, its caller ID will be matched against the pattern 1??@192.168.1.241 where "?" stands for any character. If a match is found the command *--3 will be executed.

    You can concatenate many commands in this entry using the ";" sign. For example:

    CmdBasedOnCallerID=1??@192.168.1.241=*--3;2??@192.168.1.241=*--1

- It tries to match the destination (called number) to predefined expressions that are written in the configuration file. For example:

    CmdBasedOnDestination=1??@192.168.1.241=*--3;2??@192.168.1.241=*--1

- If no matching command was found using the above techniques, SoliCall will use a the default command that is written in the configuration file:

    DefaultCmd=*--3

Log Information
In order to allow the PBXMate to create log information you need to create a directory call "log" that is the parent directory from which you run SoliCall Mate. Whenever necessary, PBXMate writes log information to a file called "errLog".

Statistical Information

If enabled in the configuration file, statistical information will be written at the end of each call. This information contains statistics on the call like: noise level, echo level, jitter, packet loss, delay etc.

6 ADVANCED CONFIGURATION

6.1 Manual activation/deactivation

The manual activation explained in this section is usually not required. You need it only if you are not activating/deactivating the PBXMate via the CLI.

To activate the pbxmate write the command:
- solicall_guard solicallpbxmate pbxmate.conf

To deactivate the pbxmate you need to:
- Do kill -9 for the solicall_guard process.
- Do kill -2 for the solicallpbxmate process. If this process is not removed, there are probably active calles. You can view the error log for more details. In that case, to force killing either use kill -9 or do kill -2 twice.

6.2 Automatic activation (for production)

This section explains how to activate the PBXMate and optionally the GUI in production. In production you let the operating system control the activation of the PBXMate.

Linux Platform

On Linux platform, you need to add the command to /etc/rc.local:

```
export SOLICALL_BIN=/usr/local/solicall/bin/
echo -e "start \n exit" | /usr/local/solicall/bin/solicall_cli
/usr/local/solicall/bin/solicall_gui -root /usr/local/solicall –ports 8083 &
```

The first command takes care of the environment variable. The second command activates CLI which in return activates the guard which in return activate the main application (solicallpbxmate). The third command is optional and it activates the GUI.

Replace the SoliCall path with the path in which you installed SoliCall on your computer.

Alternatively, you can add the above commands as a service using chkconfig.

Windows Platform

You can add the following two commands to the machine startup, usually at the registry in HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Run (on Windows 64 bit) or HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run (on Windows 32 bit).

"c:\solicall\bin\solicall_guard" "c:\solicall\bin\solicallpbxmate.exe" "c:\solicall\bin\pbxmate.conf"
"c:\solicall\bin\solicall_gui" -root "c:\solicall" -ports 8083

The first command activates the guard which in return activate the main application (solicallpbxmate). The second command is optional and it activates the GUI.
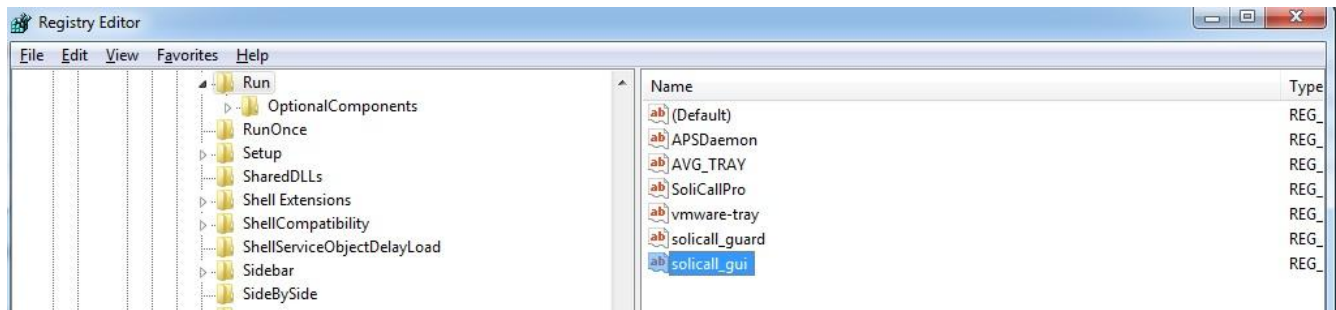
Alternatively, the PBXMate could be activated as a windows service. To do so, please ask you SoliCall's representative to receive the service version of the solicallpbxmate executable. Whenever you receive this executable, put it in the bin directory. You need to register this service via the following command:

sc create solicallpbxmate binpath= C:\SoliCall\bin\solicallpbxmate.exe start= auto DisplayName= "SoliCall PBXMate"

The service will be started every time the machine is activated. You can manually control this service via the built in tools of Microsoft (Start -> Control Panel -> Administrative Tools -> Services).

In all the above examples, replace the SoliCall path with the path in which you installed SoliCall on your computer.

6.3 Configuration File

The configuration file looks like this:
; ##################### Main Section #######################
; Config Version (version of the configuration)
; This parameter allow to upgrade the PBXMate without changing the
; configuration files/scripts.
ConfigVersion=2

; IP Version (currently only ipv4 is supported)
IPVersion=4

; Information on the SIP Proxy – usually the IP PBX.

; The information contains IP, Port (on which the PBX waits for SIP messages)
; and domain/realm to use. In case a name is provided instead of dotted IP address,
; the name will be resolved using DNS. In case the SipProxyDomain is not provided,
; The SipProxyIP will be used unless it is 127.0.0.1 and in that case SipUAIP will be used.
SipProxyIP=192.168.0.133
SipProxyPort=5060
;SipProxyDomain=internal

; Information on our machine – i.e. the machine that runs the PBXMate.
; The IP of the machine and the port to use for SIP messages.
; If you install PBXMate on the same machine as the PBX,
; or if you have sip-phone running on the same machine as the PBXMate,
; you should change the UA port to a non-standard one like 5061.
; In case a name is provided instead of dotted IP address,
; the name will be resolved using DNS.
; If the SipUAIP is not given, the PBXMate will try to auto-detect it.
;SipUAIP=192.168.0.126
SipUAPort=5061

; If you want that PBXMate to bind using a different address than SipUAIP,
; you can use the following parameter. This option can be useful,
; when you have DNAT'd network configuration.
;SipUAIPForBinding=192.168.0.126

; Following value is the initial port number for the RTP
; the program will start using ports 4000,4001, etc.
; Value should be in the range of 1024 to 49151. Do not use higher values since
; higher port numbers are dedicated for dynamic ports.
; Each phone requires 2 ports, one for RTP and one for RTCP.
; Therefore, if you allocated 6 phones (to enable 3 concurrent calls) you will
; need to make sure that ports 4000-4011 are free and not used by any other program.
; To analyze port usage on your machine you can use the command "netstat -a -n"
RtpUAPortInitVal=4000

; SoliCall Separator is the string used in order to separate
; between fields (usually the destination & filtering type) when giving a command
; to the PBXMate.
Separator=--

; The following, if exists, specifies list of commands based on the caller id.
;CmdBasedOnCallerID=1??@192.168.1.241=*--3;2??@192.168.1.241=*--1

; The following, if exists, specifies list of commands based on the destination (called number).
;CmdBasedOnDestination=1??@192.168.1.241=*--3;2??@192.168.1.241=*--1

; The following, if exists, is the default command.
; If you are using TRUNK, and each call goes to a different destination, then you
; probably want do use the following command:
DefaultCmd=*--3

; If the TrunkName exists, the PBXMate will be presented as a trunk.
; Otherwise, there is no trunk.
TrunkName=SoliCallPBXTrunk

; if RegisterTrunk is 1, the PBXMate will register the trunk with the Sip Server.
; Use 0 if you do not want to register. This parameter is not mandatory if there
; is no trunk
RegisterTrunk=1

; The value of NumCallsToFilter specifies how many concurrent channels will be filtered
; by the PBXMate. If this value is not specified the maximum that the compiled package
; can support. Please note that for production versions, this number should not exceed
; the license you are using.
;NumCallsToFilter=1

; The value of NumPhonesToAllocate specifies the number of phones
; that will be allocated. Please note that the phones work in pairs. It means
; that two phones are working together.
; If you want to filter 3 concurrent channels then you need to set this number to
; be at least 6 (= 3 x 2).
NumPhonesToAllocate=2

; If SipProxyArchitecture is 1, then PBXMate will function as a full transparent
; SIP Proxy. It will not have to store any password as it will forward all
; the authentication information from the phone to the PBX.
;SipProxyArchitecture=0

; If InviteShouldMatchPairPhoneName is 1, when an invite message is received it
; will be served only if From field or To field contain a name that appears in the pair phones
; This option is mainly used when the PBXMate is acting as Sip-Proxy for multiple phones
;InviteShouldMatchPairPhoneName=0

; OrigPhonesName lists the names of all the original phones (i.e. all the phones
; that answers the invite message).
; If this parameter does not exists, the phone names are generated automatically
; using the prefix indicated by OrigPhonesNamePrefix.
; OrigPhonesName=SoliCallOrig0;SoliCallOrig1;

; OrigPhonesSecret lists the secret of all the original phones.
; If this parameter does not exists, the value of the Secret parameter is used.
; OrigPhonesSecret=welcome0;welcome1;

; If exists, the following parameter indicates the prefix of the names
; of all automatic generated orig phones. The default is "SoliCallOrig"
; OrigPhonesNamePrefix=SoliCallOrig

; PairPhonesName lists the names of all the pair phones (i.e. all the phones

```
; that dial to the destination).
; If this parameter does not exists, the phone names are generated automatically
; using the prefix indicated by PairPhonesNamePrefix.
; PairPhonesName=SoliCallPair0;SoliCallPair1;

; PairPhonesSecret lists the secret of all the pair phones.
; If this parameter does not exists, the value of the Secret parameter is used.
; PairPhonesSecret=welcome0;welcome1;

; If exists, the following parameter indicates the prefix of the names
; of all automatic generated pair phones. The default is "SoliCallPair"
; PairPhonesNamePrefix=SoliCallPair




; if RegisterOrigPhones is 1, the PBXMate will register the original phones
; with the Sip Server. Use 0 if you do not want to register.
RegisterOrigPhones=0

; if RegisterPairPhones is 1, the PBXMate will register the pair phones with the
; Sip Server. Use 0 if you do not want to register.
RegisterPairPhones=1

; If exists the Secret is the password used for authentication. When ask for
; this password will be used for the both phones and trunk. You can override this password
; by providing specific passwords to the phones by using OrigPhonesSecret and PairPhonesSecret.
;Secret=welcome

; The RegisterExpiration is the expiration field (in seconds) for the REGISTER commands.
; If this parameter is not specified, the default is 3600 seconds.
; RegisterExpiration=3600

; The Denial of service expression, if exists, contains a list of expressions
; that represent phones for which filtering will not be done. Any Invite coming from
; these phones will result in a 503 reply.
; You can use wild char "?" to match any character.
; DOSExpressions=100@internal;2??@internal;

; The Allow of service expression, if exists, contains a list of expressions
; that represent phones that even if they exists in the DOS list will still get filtering.
; Any Invite coming from these phones will be filtered even if it appears in the DOS.
; You can use wild char "?" to match any character.
; AOSExpressions=100@internal;23?@internal;

; if StartWithFilterActivated is specified it indicates if when a call starts
; the filter is activated, or not. Default is 1. Set this parameter to 0
; only if you are not interested in filtering and you are using the PBXMate
; only to monitor the quality of the calls.
;StartWithFilterActivated=1
```

; RTPTimeoutInSeconds indicates the number of seconds of RTP/RTCP
; inactivity in any direction that will force to close the RTP session.
; Default is 30.
;RTPTimeoutInSeconds=30

; RTPTimeoutWhenOnHoldInSeconds indicates the number of seconds of RTP/RTCP
; inactivity in any direction, while the call is on Hold, that will force to close the RTP session.
; Default is 300.
;_RTPTimeoutWhenOnHoldInSeconds=300

; The following parameter let's the PBXMate know how to handle cases where there
; is more than one media (usually video in addition to audio). Many Sip devices do not
; know how to handle cases where each media has its own IP and therefore the PBXMate
; has to use a single IP for both audio & video. If your equipment do know how to handle
; such case you can change the value to "1" and save network traffic by allowing
; video packets to be routed directly between the devices without the need to
; go via the PBPXMate.
CanHaveDifferentIPForDifferentMedia=0

; If exists, the following parameter indicates if the Pair phone should impersonate
; the originator of the call. By default, this parameter is false (0) and only the display
; name of the originator of the call is being used. But on some systems this display
; name is not shown to the user and therefore the called party does not see
; who is the originator of the call. If set to true (1), the pair phone will take the full sip
; identity of the calling party to resolve the problem. This however cannot be done on
; systems where the call might be blocked by the PBX due mismatch of IP/Port etc.
; PairPhoneShouldImpersonateTheOriginator=0

; Optional sip routing file (its path is relative to SOLICALL_BIN). If exists,
; each line in this file should have the following structure:
; Sip Proxy IP::Sip Proxy Port::Expressions. For example:
;192.168.0.113::5060::100@internal;2??@internal;
; In this example, if the destination of a call is 207@internal the PBXMate will
; dial (send the INVITE message) to the sip proxy 192.168.0.113:5060
;SipRoutingFile=./pbxmate_sip_routing.conf

; ############# Noise Removal Section ###################

; Perform Noise Cancellation
NeedToDoNoiseCancellation=1

; The CPUPower is a number in the range of 0 to 10.
; The higher the number, the more CPU will be used by the algorithm
CPUPower=1

; The detect aggressive level is a number in the range of 0 to 4.
; The higher the number, the algorithm will be more aggressive in detecting the speaker.
; We recommend using the value 1
DetectAggressiveLevel=1

; The clean aggressive level is a number in the range of 5 to 10.
; The higher the number, the algorithm will be more aggressive in noise removal.
; We recommend using the value 10
CleanAggressiveLevel=10

; Set this parameter to "1" if you want SoliCall to cancel acoustic shocks – this
; feature is important mainly in systems that have low quality hardware
; that might cause spikes.
; If set to "1" the peaks in the voice of the speaker might be slightly reduced.
; Default is "0".
; CancelAcousticShock=0

; The GroupType indicates to which group type the filtering should be made.
; Value 0 means anybody. This is the default in case this parameter is not specified.
; Value 1 means adults.
; Value 2 means teenagers.
; Value 3 means children.
; GroupType=0

; If exists, the following values indicate the percentage of the original signal that will be
; generated when no speech is detected.
; By default the values are 15, 15. I.e. output 15% of the original signal.
; Alternatively you can set the values to 15, 0 to get total silence in case no voice
; is detected.
CNGInitialValue=100
CNGEndValue=100

; If exists, the following values in indicate the percentage of the original signal that will be
; generated at the end of a burst.
; By default we decrease the amplitude by 40% every second,
; and the lower value is 15%.
; BurstEndDecrease=40
BurstEndLowerValue=100

; If exists, the following value indicates if the filtering should include removal of
; noise frequencies. By default the value is 1 (i.e. true)
; RemoveNonSelfFrequencies=1

; If the NeedToCheckDTMF is 0 there will no special check
; for DTMF in the audio signal. If the value is 1 the PBXMate
; will look for DTMF in the audio signal (inband) and when found is will be

```
; passed unaltered.
NeedToCheckDTMF=1

; If exists, it indicates is to bypass the VAD operation. By default the value is 0
;(i.e. false). Bypassing VAD can increase performance.
; If you set this value to true, you must set:  CNGEndValue to 100
BypassVAD=0

; ############# Echo Cancellation (AEC) Section ##################

; Perform Echo Cancellation (AEC)
NeedToDoAEC=0

; AEC Algorithm. Can be either advanced or basic
; For professional tuning the following types could also be used: advanced-4 and advanced-7
AECAlgorithm=advanced

; The following parameter, if exists, defines the aggressiveness of the echo cancellation.
; The higher the number the more aggressive the algorithm will be. Values can range from
; 50 to 300. Default is 100.
MaxCoefInAECParam=100

; The following parameter, if exists, defines the tail of the echo. Values can range from
; 0 to 5. Default is 1. The higher this number the algorithm will look for longer tail.
AECTailType=1

; The following parameter, if exists, defines the suppression done on echo by
; the "advanced" algorithm. It can range from 0 to 100.
; The lower the value the more aggressive cancellation will be.
AECMinOutputPercentageDuringEcho=0

; The following parameter, if exists, defines the treatment level in howling.
; The range of values is between 0 to 20. Default is 10.
; The higher the value the detection/treatment in howling is more aggressive.
;AECHowlingLevel=10

; ############# Automatic Gain Control (AGC) Section ##################

; Perform AGC (by default disabled). In order to activate AGC, Noise cancellation
; must also be enabled (NeedToDoNoiseCancellation=1) and it is recommended
; that the BypassVAD will be disabled (BypassVAD =0).
NeedToDoAGC=1


; If AGC is activated the level should be a number between 0 to 64000
AGCLevel=32000

; ################### Statistics Section ##################
; If StatisticsFile is specified, the PBXMate will add statistics to this file.
```

; the path is relative to SOLICALL_BIN
StatisticsFile=../log/stats

; If exists, the following define the threshold for highlight the Noise Level. Default 3000
;ThresholdToHighlightNoiseLevel=3000

; If exists, the following define the threshold for highlight the AGC Coef. Default 300
;ThresholdToHighlightAGCCoef=300

; If exists, the following define the threshold for highlight the Echo Level. Default 1000
;ThresholdToHighlightEchoLevel=1000

; If exists, the following define the threshold for highlight the jitter. Default 10 (ms)
;ThresholdToHighlightJitter=10

; If exists, the following define the threshold for highlight the delay. Default 100 (ms)
;ThresholdToHighlightDelay=100

; If exists, the following define the threshold for highlight the packet loss. Default 0.1 (percent)
;ThresholdToHighlightPacketLoss=0.1

; If exists, the following define the threshold for highlight the MOS. Default 3.6
;ThresholdToHighlightMOS=3.6

; If the following files are specified, the PBXMate will write real time statistics
; on active calls to these HTML file. You can browse this file via any browser.
; the path is relative to SOLICALL_BIN
RealTimeAverageStatisticsFile=../log/RealTimeAverageStats.html
RealTimeExtremeStatisticsFile=../log/RealTimeExtremeStats.html

; If exists it specifies the refresh rate (in seconds) of the real-time statistics.
; Default is 10 seconds.
;RefreshRateOfRealTimeStatistics=10

; ################### DTMF Section ###################
; The following group of parameters (if exists) are the DTMF controls for SoliCall.
; SoliCall can detect both inband DTMF (assuming the NeedToCheckDTMF
; parameter is set) or via rfc2833.
; For example let's assume:
; DTMFRegistraterPrefix=#7
; DTMFActivatePrefix=#9
; DTMFSuffix=#
; DTMFDisable=*9
; 1) In order to start registration the speaker needs to press:
; <DTMFRegistraterPrefix><PID=Personal ID><DTMFSuffix>
; For example: "#7 12345 #"   (while 12345 is the PID).

; Then the speaker needs to say few sentences (from a quiet area without
; background noise). When the call is completed, the registration data will be saved
; in the Registration Library.
; 2) In order to activate a personalized filtering (a.k.a. My Voice Only) during a call,
; the speaker needs to press:
; <DTMFActivatePrefix><PID><DTMFSuffix>
; For example: "#9 12345 #"
;3) In order to activate regular filtering (a.k.a. Voice Only), for both directions, during a call,
; the speaker needs to press:
; <DTMFActivatePrefix><DTMFSuffix>
; For example: "#9 #"
;4) In order to disable filtering (for both directions) during a call,
; the speaker needs to press:
; <DTMFDisable>
; For example: "*9"
; The DTMFOKToneFile & DTMFFailToneFile should point to RAW PCM 8khz
; 2 bytes per sample short audio files to be played to the speaker on successful
; command or failure.
;DTMFRegistraterPrefix=#7
DTMFActivatePrefix=#9
DTMFSuffix=#
DTMFDisable=*9
; the following paths is relative to SOLICALL_BIN
;RegistrationLibrary=../registration/
DTMFOKToneFile=./okTone.raw
;DTMFFailToneFile=./failTone.raw

; #################### Misc Section ####################
; If exists, the original calls (i.e. before filtering) are recorded in the indicated library.
; the path is relative to SOLICALL_BIN.
; The format of the recorded files is RAW, 16 bits per sample.
; 8Khz if narrowband codec is used, 16khz if wideband codec is used.
; You can convert these files to wav using the sox utility or contact us to get
; a utility to reconstruct wav files from the raw data.
;OrigRecordingLibrary=../rec/

; If exists, the clean calls (i.e. after filtering) are recorded in the indicated library.
; the path is relative to SOLICALL_BIN.
; The format is the same as for the original calls.
;CleanRecordingLibrary=../rec/

; If set to 1, All sip messages will be printed to the errLog
;SipDebug=0

; For non-evaluation package you need to supply license file
; the path is relative to SOLICALL_BIN

;LicenseFile=./mylicense.lic

6.4 Tips & Tricks for configuration

Noise Reduction Parameters

The following specify different options for noise reduction. They are ordered from the least aggressive option to the most aggressive noise cancellation.

Filtering audio coming from Conference Rooms
If you are filtering audio coming from conference rooms, set the following values:
CNGInitialValue=100
CNGEndValue = 100
BurstEndLowerValue = 100

Also, if you use these parameters, you can also reduce CPU requirements by setting:
CPUPower=0.
If you are not interested in AGC and you need to reduce CPU consumption you can also set:
BypassVAD=1.

Filtering Calls with High Level of Noise
Otherwise, If you filter audio coming from environment with constant <u>big noise</u>, set the following values:
CNGInitialValue=15
CNGEndValue = 15
BurstEndLowerValue = 40

All other cases
Otherwise, set the following values:
CNGInitialValue=15
CNGEndValue = 15
BurstEndLowerValue = 15

In addition to enhance the noise reduction you can use the personalization option. A detailed explanation of how to use it can be found in the section describing the DTMF controls in the configuration file.

Echo Cancellation Parameters

PBXMate has two different types of AEC algorithm (control by the parameter AECAlgorithm):
- Basic – Low CPU, basic algorithm.
- Advanced – Complex algorithm with advanced self tuning.

If you suffer from very long delays, try to increase the value of AECTailType.

If you suffer from high volume echo, try to increase the value of the parameter: MaxCoefInAECParam.

Filtering Conference Calls

In case the PBXMate is used to improve conference calls, configure the PBXMate to filter the audio entering the conference bridge:

```
DefaultCmd=*--1
```

The above instructs the PBXMate to send all calls to their original destination and to filter the audio coming from the originator. To save CPU time and also to improve quality, we do not filter the mixed audio coming back from the conference bridge. Only the audio entering the conference bridge should be filtered.

6.5 Integration with H.323 network

The PBXMate can integrate with H.323 network via a mediator that translates between SIP to H.323 signaling. There are several products that can do this work. SoliCall's lab has successfully tested the following configuration:
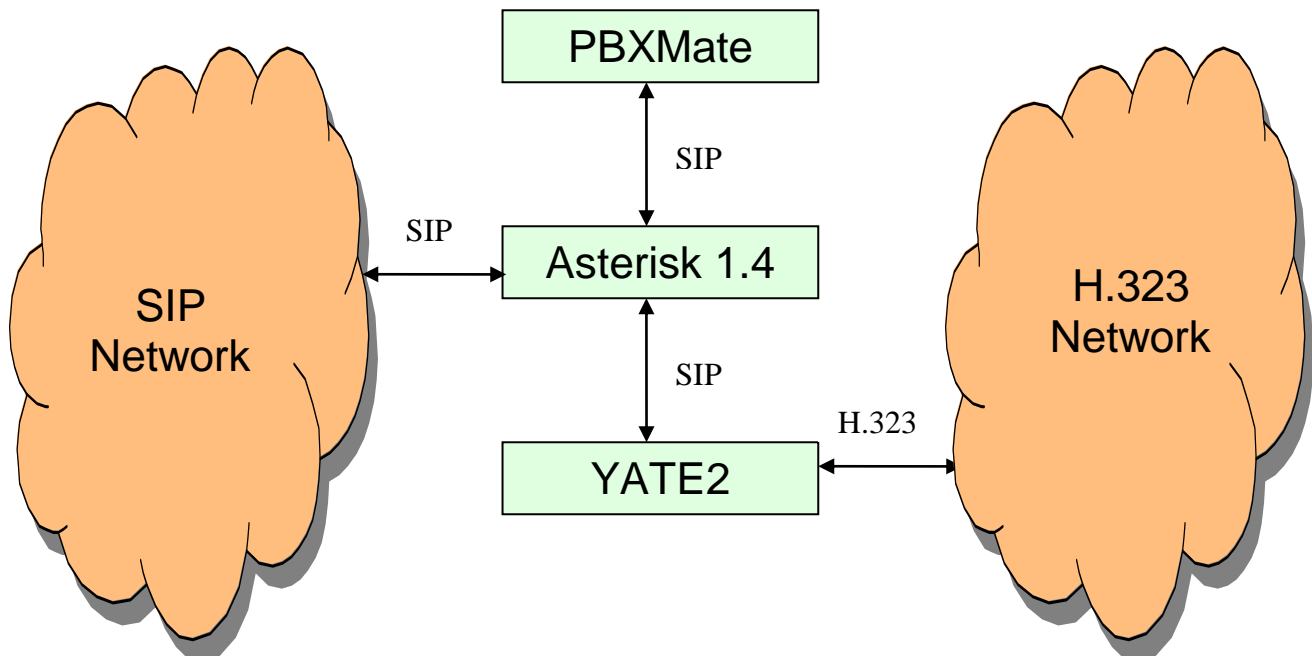
Yate Configuration

Yate configuration is straightforward based on its published setup. Please refer to
http://yate.null.ro/pmwiki/index.php?n=Main.H323ToSIPSignallingProxy.

H323chan.conf

```
[general]
external_rtp=yes
passtrough_rtp=yes  ; note the incorrect spelling

[codecs]
default=no
mulaw=yes
alaw=yes

[ep]
faststart=on
```

ysipchan.conf

```
[codecs]
default=off
mulaw=yes
alaw=yes
```

regexroute.conf:

```
[default]
${rtp_forward}possible=;rtp_forward=yes
${formats}^\([^,]*\)=;formats=\1
${module}^sip$=h323/${called}@192.168.0.166
${module}^h323$=sip/sip:${called}@192.168.0.113
.*=-;error=forbidden;reason=Protocol not allowed
```

Asterisk Configuration

Asterisk configuration is very simple. In our lab we tested the following setup which diverts every SIP generated call to H.323 network and every H.323 generated call to SIP network.

Divert all SIP generated calls to PBXMate and afterwards to the Yate Server

```
[sip-generated-calls]
exten => _X!,1,GotoIf($["${SIPUSERAGENT:0:8}" = "SoliCall"]?3:2)
exten => _X!,2,Dial(SIP/SoliCallPBXTrunk/${EXTEN})
exten => _X!,3,Dial(SIP/${EXTEN}@192.168.0.126)
```

Divert all H.323 generated calls to PBXMate and afterwards to their original destination.

```
[h323-generated-calls]
exten => _X!,1,GotoIf($["${SIPUSERAGENT:0:8}" = "SoliCall"]?3:2)
exten => _X!,2,Dial(SIP/SoliCallPBXTrunk/${EXTEN})
exten => _X!,3,Dial(SIP/${EXTEN})
```

6.6 Multi instances on a single computer

If required, you can contact SoliCall support and request to get a special version of the PBXMate that supports activation of multiple instances on a single computer. This configuration might be beneficial in complex environments. For example:

- In case you want to run on a single computer two trunks of PBXMate – each one connected to a different PBX.
- In case you want to use more than one configuration setup.

Preparing the configuration files

For every instance you should prepare a dedicated configuration file:

- pbxmate.conf – configuration file for the first instance.
- pbxmateB.conf – configuration file for the second instance.
- pbxmateC.conf – configuration file for the third instance.

Following are the recommended changes for each new configuration file:

- SipUAPort - there should be a unique number for each instance.

- RtpUAPortInitVal should be modified to avoid collision. For example if a single instance of PBXMate you are using can support up to 200 calls, it will require 800 ports (200x4). Therefore if for the first instance RtpUAPortInitVal is set to 4000, it will use port numbers from 4000 to 4799 (in a random order). As a result, for the second instance you should configure RtpUAPortInitVal to be 4800 or higher.

- Statistics file should be unique for each instance. For example for the second instance you might configure:
  - StatisticsFile=../log/statsB
  - RealTimeAverageStatisticsFile=../log/RealTimeAverageStatsB.html
  - RealTimeExtremeStatisticsFile=../log/RealTimeExtremeStatsB.html

- Modify SIP names for the trunk & phones to be unique. For example for the second instance you might configure:
  - TrunkName=SoliCallPBXTrunkB
  - OrigPhonesNamePrefix=SoliCallOrigB
  - PairPhonesNamePrefix=SoliCallPairB

Log files

Each instance will automatically write to a different log file. For example, the second instance will write errors to the file errLogB.

Activation via CLI

For the first instance, you use the regular commands like:
- start
- stop
- status
- show_bindings

For the second instance, you add "B" after the name of the command like:
- start B
- stop B
- status B
- show_bindings B

For the third instance, you add "C" after the name of the command like:
- start C
- stop C
- status C
- show_bindings C

Manual Activation

To activate the first instance, write as before:
- solicall_guard solicallpbxmate pbxmate.conf

To activate the second instance, write:
- solicall_guard solicallpbxmate pbxmateB.conf B

6.7 Professional Tuning of Algorithms

Since the algorithms have many tuning options, our support team will be happy to help you tune the algorithms for best performance in your system. To use this option please do the following:

Activate Recording

The PBXMate has built-in recording capability of the audio signal "before" & "after" including synchronization information. To activate this capability uncomment, in the file pbxmate.conf, the lines
OrigRecordingLibrary=../rec/
CleanRecordingLibrary=../rec/

Afterwards, restart PBXMate. If you have done it correctly, every call you make will now be recorded and stored in the sub-directory "rec". Few files will be generated for each call.

Make Short Test Calls

Make 1-3 short calls, 1-2 minutes per call is enough.

Send Recording

Zip the recorded files that were created and e-mail to SoliCall support. The recordings will be analyzed and you will receive tuning instructions for best performance in your system.

7 <u>FAQ</u>

7.1 Basic Configuration

Which Port number to use for SIP?

If you run the PBXMate on the same machine as the PBX or in case you want to run a soft-phone on the same machine, you should modify the UA to use a non-standard port (e.g. SipUAPort=5061). Otherwise, you can continue using the standard port (5060).

Can the PBXMate work with H.323?

Yes. Details configuration for this architecture appear in this manual.

Will the PBXMate work with my PBX type?

The PBXMate is working in production with many types of PBX. We do not monitor the exact brand/type of each PBX since it is of no importance as the PBXMate communicate with the PBX using <u>standard protocols</u>. Therefore, if your PBX complies with SIP or H.323 standards, there should be no problem.

Is there and automatic way to reload configuration parameters?

The PBXMate has an option reload the configuration file during run time. Please note that this reload operation will not affect network/port parameters like IP address, port etc. In order to do so, send the following UDP message to the SIP socket that the PBXMate is listening to:
> 800-PBXMATE-RELOAD C:\Program Files (x86)\SoliCall\bin\pbxmate.conf\r\n\r\n

7.2 SIP Registration

Who is doing registration in case of Sip Proxy Architecture?

When you use the PBXMate is used in Sip Proxy Architecture, the PBXMate routes to the PBX all registration requests coming to it from the phones. The PBXMate does not independently initiate any registration request.

Who is doing registration?

By default (Sip Trunk Architecture), the PBXMate tries to register, with your PBX, a SIP trunk called SoliCallPBXTrunk and a single pair phone called SoliCallPair0. If required, the PBXMate also accepts any registration request coming from your PBX.

Do I need to register the Pair phones?

By default (Sip Trunk Architecture), the pair phones are initiating calls. If your PBX requires that phones that initiate call are registered, then you must register the pair phones in your PBX. Otherwise, you can disable their registration by changing in the configuration file RegisterPairPhones=0.

Does PBXMate register with password?

Yes, PBXMate has the option of using a password for registration (secret). The username that is used is the SIP names (of the phones & trunk).

7.3 Call Initiation

When making a call the caller ID is SoliCallPair0 and not the original Caller ID.

This problem might happen only in Sip Trunk architecture and when your PBX is configured to show the sip name and not the display name.

Solution: Allow SoliCall Pair phones to impersonate the originator by change *PairPhoneShouldImpersonateTheOriginator*=1

7.4 Call Volume

How can I modify the number of calls the PBXMate can handle concurrently?

This is controlled in the configuration file by the parameter *NumCallsToFilter*.

What is the limit for number of concurrent calls?

The evaluation/demo package has a built-in limit of few (3) concurrent calls. The production version can handle hundreds of concurrent calls.

What happens when the PBXMate reaches its limit for number of concurrent calls?

When the PBXMate reaches its limit for number of concurrent channels, any new call will receive a response of "486 (Busy Here)". In Sip Trunk architecture, the dial-plan in the PBX should be configured to route the call directly to its original destination.

Can I install multiple instances of PBXMate?

Yes. You can install multiple instances of PBXMate on the same computer and on different computers. Please see chapter 6.6 for more details.

7.5 Call Statistics

How can I see call statistics?

There is a graphical real time view that shows statistics on all active call. To view this presentation just open with your browser the files "RealTimeAverageStats.html" and "RealTimeExtremeStats.html" which are located in the log directory. The first file shows average values during the call while the second shows extreme values that where monitored in the call.

In addition, the statistics of all the calls is stored in the log directory under a file name "stats".

Which statistic data appears in the stats file?

For every call that is ends, the stats file contains a line with statistics.

The content of the line is as follows:
- Date, Time & length of the call.
- For each of the two audio direction:
    - Confident Voice – percent of time the PBXMate identified a voice with high confidence.
    - Noise Level – average & max noise level during the call.
    - AGC Coef – the average and max AGC coefficient during the call. For example, the number 300 means that the AGC algorithm had to amplify the original audio three times.
    - Echo Level – average & max echo level during the call.
    - Jitter – average & max jitter in ms.
    - Packet Loss – percent of packet loss.
    - Delay – average & max delay in ms.
    - MOS – average & min score calculated based on jitter, packet loss & delay.

Which statistic data appears in the graphical view?

The graphical view contains similar information to the one appearing in the statistics file. It also highlights problematic numbers. You can control the threshold via the configuration file.

In the graphical view I do not see the Icon of SoliCall in the header. Why?

If you change the location of the html file, you need to make sure you have in the same location both the icon (Win2000_16.ico) & Flash subdirectory. You can copy these files from the log directory.

Noise Level is always zero. Why?

If the PBXMate is **not** configured to do noise cancellation, it is not monitoring the noise in the call. You need to enable noise cancellation in order to monitor the noise level.
NeedToDoNoiseCancellation=1

AGC Coefficient is always 100. Why?

If the PBXMate is **not** configured to do AGC, it is not monitoring the AGC in the call. You need to enable AGC in order to monitor the AGC Coef.

`NeedToDoAGC=1`

Echo Level is always zero. Why?

If the PBXMate is **not** configured to do echo cancellation, it is not monitoring the echo in the call. You need to enable echo cancellation in order to monitor the echo level.

`NeedToDoAEC=1`

Confident Voice is always 100%. Why?

If the PBXMate is **not** configured to do noise cancellation, it is not monitoring the voice in the call. You need to enable noise cancellation in order to monitor the voice.

`NeedToDoNoiseCancellation=1`

Only one of the two audio directions contains real information. Why?

Statistics is generated only for the direction that was filtered (based on the filtering type).
If you want to have statistics for both directions, you need to use filtering type to be "3".
Alternatively, if echo cancellation is activated, the PBXMate has to analyze both directions of the call and it that case it also generated statistics for both directions.

How to I monitor statistics without modifying the audio?

In the rare cases you do not want to do any change to the audio but just monitor the quality, you can configure the PBXMate not do change the audio:

`StartWithFilterActivated=0`

7.6 SoliCall GUI

How was the graphical interface built ?

The graphical interface was built using mongoose web-server engine and for advanced users, per request we will send you the source code of solicall_gui.

Can I change the port used by solicall_gui ?

You can change the port used by solicall_gui via the ports parameter. For example: solicall_gui -ports 8080

7.7 Licensing Model

What is the licensing model for the PBXMate?

The PBXMate is licensed per number of concurrent channels. The license of the PBXMate is node-lock. When you purchase a license for the PBXMate, a key is generated for you based on the MAC address of one of the network cards on your destination machine.

Is the term "number of concurrent channels" identical to "number of concurrent calls"?

Usually this is true except for the case of internal calls whenever the PBXMate is used in Sip Proxy Architecture. I such scenario, a single internal call will pass via the PBXMate twice and therefore will consume two of its channels.

Does call waiting being counted?

Every call that passes via PBXMate consumes one of its channels even if the audio in this call did not start yet.

What information should I provide to receive a production version?

The PBXMate evaluation version that you are using, can automatically handle it. It will send the necessary information to enable generating license key and receive the matching production version.

Alternatively, you can manually send the following information:

On Linux
- Send the output of the command "/sbin/ifconfig" or the file "errLog".
- Send the output of the commands "cat /etc/*release*" and "uname –a"

On Windows
- Send the output of the command "ipconfig /all"
- Specify that you are using Windows architecture.