

How to solve audio issues caused by Asterisk timing configuration in VoipNow

Applies to VoipNow 3 and higher!

The current article describes various timing modules and explains how to eliminate audio issues triggered by the way Asterisk timing is configured in VoipNow.

What is timing

Timing affects the RTP flow in calls that use RTP (SIP, H.323, MGCP). Since the RTP system is normally clocked on incoming packets, in some situations you may reach a **standstill**. That's when internal timing takes over, forcing Asterisk to send packets based on the loaded module and not on the incoming flow.

A **standstill** occurs when both systems are waiting for a packet to arrive and neither of them is sending anything.

Timing interfaces

On older Asterisk versions (1.4.x), if the Asterisk system required internal timing, the only source acceptable was DAHDI. Starting with Asterisk 1.6.1, there is a new timing API, which allows various timing modules to be used.

Here are the timing modules included in Asterisk:

res_timing_dahdi

- uses timing mechanisms provided by DAHDI;
- used to be the only means by which Asterisk could receive timing;
- can be very efficient, especially if the system is going to use DAHDI hardware;
- may not be so attractive if DAHDI is strictly used as a timing source;

res_timing_timerfd

- is a relatively new feature on Linux, which means that it may not be widely spread;
- is only available on Linux systems using a kernel version that is at least 2.6.25 and a glibc version that is at least 2.8;
- starting with Asterisk 1.6.2.x uses a timing mechanism provided directly by the Linux kernel;

res_timing_pthread

- uses the POSIX pthreads library in order to provide timing;
- is portable to many types of systems because the code uses a commonly-implemented set of functions;
- is the only timing source, currently usable on a non-Linux system;
- is also the least efficient timing source, known for losing its effectiveness in a heavily-loaded environment, because a single user-space thread is used to provide timing for all users of the timer;

VoipNow behavior

By default, in VoipNow, Asterisk builds and loads all the timing interfaces set in the `/etc/asterisk/modules.conf` file; `res_timing_pthread` is the first module to get loaded and it is followed by `res_timing_dahdi`;

If you want to change timing, you need to edit the `modules.conf` file as explained below:

Step 1: Replace the `modules.conf`.

```
Replace:

[modules]
preload => res_timing_pthread.so
preload => res_timing_dahdi.so

With:

[modules]
preload => res_timing_dahdi.so
preload => res_timing_pthread.so
```

Step 2: Restart Asterisk.

A piece of advice

To better understand how Asterisk timing works, it would probably be best to read more about it [here](#).

However, if you already used Dahdi timing interfaces and didn't encounter any timing issues, it is better to stick to this approach. If not, we recommend that you use the `res_timing_timerfd` interface.

If using `res_timing_dahdi` and `res_timing_timerfd` leads to timing bugs, it is recommended that you try the `res_timing_pthread` interface.

Resources

[Asterisk Wiki](#)
[Asterisk Doc](#)

Related articles

- [Troubleshooting fax session failures caused by incorrect negotiation of T38FaxMaxDatagram parameter](#)
- [Troubleshooting choppy sound in MOH files played over the phone](#)
- [Troubleshooting calls and debug steps](#)
- [How to use SIP trunking to connect a PBX to an extension](#)
- [How to debug Asterisk and Kamailio](#)