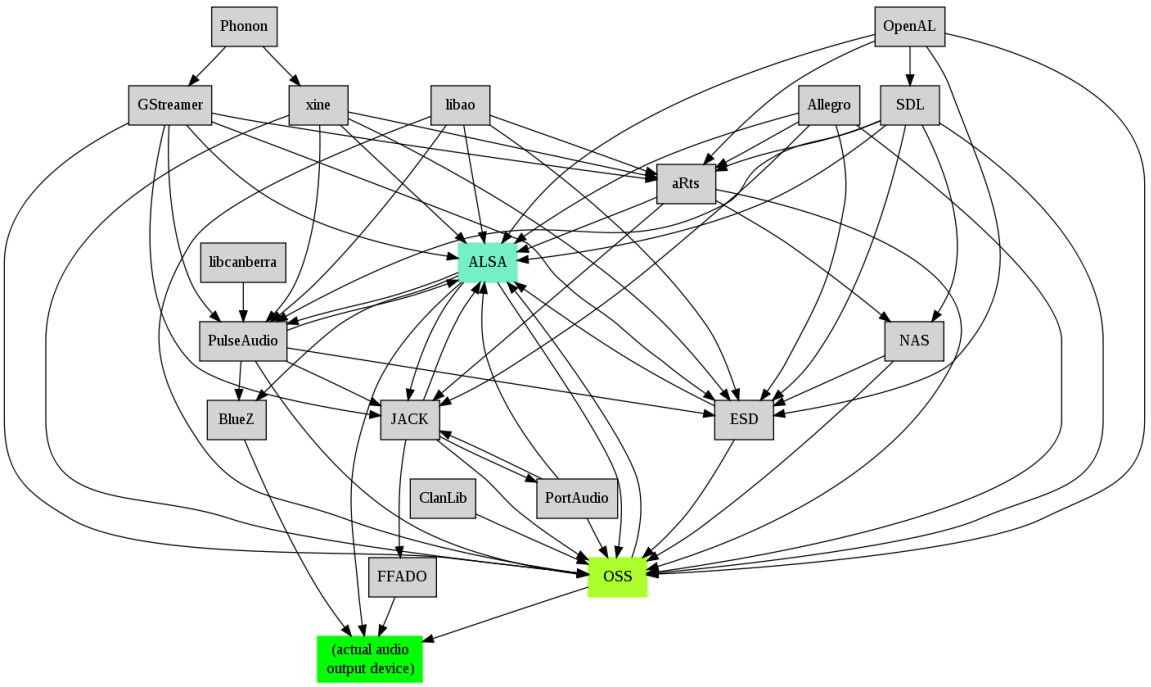
How to solve ALSA sound problems with old Linux programs and games depending on (OSS)'s /dev/dsp / fix wine games and pulseaudio problems - My few thoughts on OSS and ALSA

Author : admin



The Linux Audio Mess
Origin: Mike Melanson, http://blogs.adobe.com/penguin.swf/
Updated October 10, 2008

I remember **GNU / Linux**, 11 years from now, times when ALSA was not standardly shipped with Linux. Back then *ALSA* still lacked good support for many SoundCards and was still a "baby project".
In that time what we used to have sound on Linux was **OSS** - *Open Sound System*. *OSS* emerged right after the first ever Linux sound system **VoxWare** (formerly known as the Linux Sound Driver).

Back in those days *OSS* was used for multimedia support on both GNU / Linux and BSD based free OSes. It was few years later when I heard and used ALSA for a fist time and it wasn't really a love from first sigth.

One can easily find out by the name ALSA it is a system especially built for the Linux kernel and that's one of the reasosn why *BSD systems has their custom separate sound system.
There is plenty of reasons why OSS was substituted with ALSA. Main reason was its commercial like license, OSS wasn't completely "open source" GPLed (free software), there was resctions on use of OSS for commercial goals.

With its emerge *ALSA* started to push away OSS slowly. Somewhere in 2003, alsa has officially entered the Linux kernel source and until 2005 it was the default standard for all GNU / Linux operating systems.

As of time of writting ALSA has become the only sound system to have support for multiple sound card devices for Linux.
My experiences with ALSA, however ain't so nice if I take a look in my past experiences.
Since the very beginning of using ALSA, I had plenty of troubles with configuring properly my sound card not to mention, even after configuring it the MIDI support was not there.
Besides all the troubles main problems were stemming from the many applications still written to use OSS as sound system and hence with those sound was impossible with ALSA.The most problematic thing about apps written with OSS in mind was all of them tried to stream sound via **/dev/dsp** (OSS Digital Sound Processor), since alsa did not used /dev/dsp those programs was soundless.

On the other hand OSS was creating issues as well, one severe problem with OSS was the inability to stream multiple sounds simultaneously, because each sound stream required to pass voice through */dev/dsp* and usually there was only one /dev/dsp.

The message;

/dev/dsp: Device or resource busy
and the proceeding irritation that used to annoy us in the early GNU / Linux days had of course some raw workarounds hacks but generally the workaround did not fix problems always.

Introduction of alsa free us from **/dev/dsp** issues but on the other handy has created a whole ocean of new BIG problems ...
ALSA has modular structure and this imposes a great problem nowdays. The modular architecture is generally a good idea, however the way this was implemented within ALSA is far away from clear and easy to understand by the end user and therefore makes it very unintuitive and obscure.
Alsa misses simplicity which somehow was partially in the days of OSS. Thinking over the general situation with Linux multimedia nowdays, I believe it was exactly ALSA Project responsible for the so delayed mass Desktop Linux adoption.

Many long year standing Linux users had certainly had the *alsa* troubles during new system installs (correct me if I'm wrong).
The only fix to multiple soundcard initialization problems was to download alsa source and compile from source and hence made it hard and discouraging for people giving Linux a try.
This kind of ALSA "brokenness" pattern continues even to this very day (in Debian) Linux and probably building the alsa system from source is among the good practices to have a functional Linux sound system...

With all said the historic reason why ALSA was not quickly adopted and still is not a preferred default system for many applications ported to Free Software OSes by commercial company vendors is clear. Its simply not working out of the box ...

Hope some ALSA developers will read this post work on changing the crazy structure of ALSA over complexity. ALSA needs automate way to solve issues with itself, the configuration should be more trivial and unified if Linux has to become more attractive for Desktop adoption.

Anyways, after the few words of history and indicating my pesonal observations on ALSA. I will proceed

and explain few things on **how ALSA can be configured to support and play nice with OSS dependant programs as well some basic explanations on common incompatibility between esd and pulseaudio and how this can be fixed;**.

To assure nowdays OSS API built programs and games would work with Alsa its necessery to have installed;

ALSA wrapper for OSS applications

On Debian, Ubuntu, Fedora and most Linux distributions the *Alsa OSS compatability layer* comes under a (deb / rpm) package named **alsa-oss**

To install OSS compatability on Debian, Ubuntu and the like Debian based distributions issue:

debian:~# apt-get install alsa-oss alsaplayer-oss
...

On *Fedora* and other rpm based distributions install is with:

[root@fedora ~]# yum install alsa-oss alsaplayer-oss
...

alsa-oss provides with a command called **aoss** that should be used to work around some issues with old applications still depending on OSS:

hipo@debian:~$ aoss programName

Using aoss is helpful especially in situations if you have to run programs which deal with MIDI and others which somehow want to use **/dev/dsp**

There is also alternative way to enable alsa native support for MIDI and OSS by loading 3 kernel modules:

debian:~# modprobe snd-seq-oss
debian:~# modprobe snd-pcm-oss
debian:~# modprobe snd-mixer-oss

Note! The three modules has to be separately build using kernel source at most cases and does not come with most Linux distributions, so on many installations (including my current), they will be missing. If for you they load properly or you have customly build them add them also to load on system boot, like so:

echo 'snd-seq-oss' >> /etc/modules

```
echo 'snd-pcm-oss' >> /etc/modules
echo 'snd-mixer-oss' >> /etc/modules
```

The Linux sound situation becomes even more messy when **ESD** enters the scene. Many of the novice new Linux users certainly don't remember **(Enlightened Sound Daemon)** . *ESD* historically preceded *PulseAudio* . Hence it will be good to mention ESD was used for few years in GNOME and in around 2006-2007 it was substituted by *PulseAudio*.
Many applications, however who was ported or written for Linux especially (the proprietary ported ones) was already built to work with ESD and even though newer GNOME releases was fully using pulseaudio, this (non free software apps and games) were still depending on ESD.

The situation was partially fixed by creation of module for pulseaudio which added emulation support for *esd* . This was done by a module library for pulseaudio called **libprotocol-esound.so**
The package for almost all Linux distributions which does the esd emulation via pulse is **pulseaudio-esound-compat** . In latest Fedora Linux *pulseaudio-esound-compat* is installed by default.
In Debian and other Linux distributions it might need to be installed via apt with;

```
debian:~# apt-get install pulseaudio-esound-compat
...
```

pulseaudio-esound-compat solves some of the ESD app incompability but not always ...
Handy tool also worthy to mention in solving PulseAudio, OSS incompatibility issues is *padsp*

padsp is helpful in solving obsolete issues with OSS applications (trying to access /dev/dsp) and therefore unable to communicate with **Pulseaudio**
padsp - is a PulseAudio OSS Wrapper.

An example where *padsp* is helpful is in case of */dev/dsp* errors like:

/dev/dsp: Device or resource busy
Could not open /dev/dsp

Another common problem with sound on Linux is when running windows applications (running windows games with wine).
Quite often sound fails to work since wine tries to directly communicate with alsa and fails because alsa sound channel is taken by pulseaudio.

To workaround wine issues with pulseaudio, one of the solutions is to temporary stop pulseaudio, before running the wine emulated application:

```
hipo@debian:~$ pulseaudio --kill
```

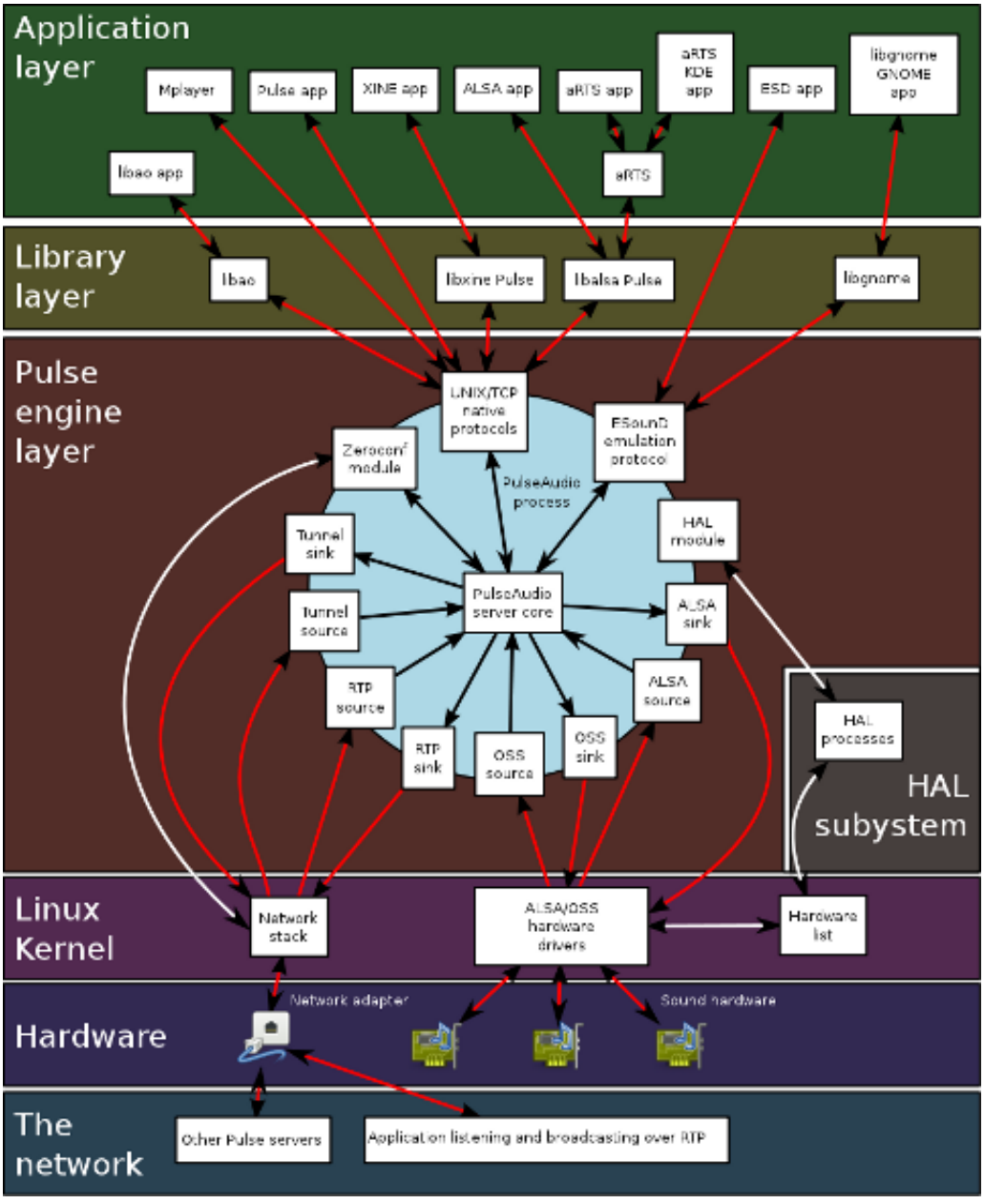
Later on when the windows *wine* emulation is completed, pulseaudio has to be started once again in

order to make Pulseaudio applications produce sound again, e.g. one has to issue:

hipo@debian:~$ pulseaudio --start
Alternative way to workaround wine sound issues is by using a script to kill pulseaudio every second. Here is fix_pulseaudio_wine_sound_probs.sh script

This script was reported by many people as **fix to problems with wine games failing to play sounds and music**, anyhow I personally prefer using the stop / start pulseaudio method.

The picture below is taken from Wikipedia and illustrates, clearly the intergalactical complexity of sound systems on Gnu / Linux and BSD



I just hope one day this (OSS, ALSA, esd, Pulseaudio) mess will be over! In the mean time I hope my

suggested work arounds helps someone. If someone has a better more unified script or solution please share in comments