

# Processing for Streaming vs. On-Air

Jeff Keith on why we can't paint streaming with the same audio processing brush

By [Jeff Keith](#) .

Published: September 9, 2020

*The author is senior product development engineer, audio processing for Wheatstone.*

We now know that audio processing is needed for streaming, and for many of the same reasons that processing is needed for on-air. We also know that we can't paint streaming with the same audio processing brush.

Here is why:

**Fast time constants can interfere with the codec.** The conventional approach of applying multiband gain control followed by fast compression to build uniform loudness and density from one music source to the next works beautifully for on-air, but it doesn't work for streaming. This is because applying fast compression time constants increases the intermodulation and other distortion products that causes codecs to make mistakes and remove or add frequencies

that it shouldn't.

That can be bad for any stream, but it's especially bad for low-bitrate streams that don't have a lot of data bits to begin with. Processors designed for streaming applications use adaptive algorithms and other less extreme measures to create uniform loudness between songs.

**Peak Overshoot is a problem.** Unlike analog, digital audio gets ugly once it tries to go past 0 dBFS, the point at which there are simply no more bits left and nasty distortion ensues.

The recommended peak input level for most codecs is around  $-3$  dBFS so a limiter is necessary to ensure that level is never exceeded. But not any old limiter will do. Aggressive limiting and its byproducts can be problematic because codecs can multiply the audibility of limiting to the point of being objectionable, and often at the expense of removing frequencies that add to the quality of music.

A good streaming processor will avoid aggressive limiting altogether. This is why in the case of StreamBlade, we designed the processor to anticipate overshoots earlier in the processing stages and designed specialized final limiters that don't add the program density that can set off issues with the codec.

**Nope. Just nope.** We learned quickly that clipping is not a good idea for peak control of program streams because it creates harmonics that aren't in the original program and because the encoder doesn't know what to do with that. In some cases, it throws bits at the bad harmonics, so you actually get more ... bad harmonics.

Clipping byproducts can sound much, much worse once a codec gets a hold of them. The good news is that streaming doesn't need the pre-emphasis can of worms that got FM broadcasters into heavy clipping to be competitive, so clipping isn't necessary or even desired for keeping peak levels in check.

**Stereo, not so much.** Big swings in L-R can trick certain codecs into disproportionately encoding stereo energy rather than more "up front" and audible program content. We experimented with various codecs and bitrates in our lab and in the field, and we found that stereo can be applied in most cases if it's done consistently and without the big L-R swings that skew the codec algorithm in favor of L-R over original content. The exception is extremely low bitrates, in which case mono is preferable.



Compression too fast for program content



Adaptive Compression - Just Right!

*The picture at left represents what happens if you use aggressive compression on encoded audio similar to that used in FM processing. Extreme time constants create distortion that not only mask subtle detail but cause the encoder to allocate bits to that distortion instead of to the desired program content. Less detail makes the audio sound "blurry." Added distortion is bad for any stream but it's especially bad for low-bitrate streams that end up with less of the original audio as it is.*

*Comment on this or any article to [radioworld@futurenet.com](mailto:radioworld@futurenet.com).*

*The author is Wheatstone's audio processing senior product development engineer and was lead development engineer for the company's StreamBlade audio processor and WheatNet-IP audio network appliance.*

## **Subscribe**

For more stories like this, and to keep up to date with all our market leading news, features and analysis, sign up to our newsletter [here](#).