– The Technical Case for Web Stream Processing –

By Jeffrey A. Keith, CPBE, NCE
Senior Product Development Engineer
Wheatstone Corporation
December 2011

INTRODUCTION

Recent statistics on the growth of online listening revealed that if broadcasters aren't yet tuned in to web streaming technology and using it to distribute their programming to fixed and mobile listeners, they could be missing out on a new form of audience *and* new revenue. Many stations now realize the importance of streaming their programming to the Internet, and for most it's become a mission critical part of their every day business.

NEW REVENUE!

According to a 2011 study by Arbitron and Edison Media, an estimated 89 million Americans listened to online radio in recent months. That same study also revealed that about 56% of the listeners tuning into a web stream did it to hear the stream of a *commercially licensed* radio station. It wasn't clear from the survey if those listeners could have tuned into a station's over-the-air signal instead. What *was* clear however, was that the audience for online listening – an audience that was only 69 million strong a year ago – has been *doubling* in size every 5 years since data mining on online listening habits began in 2001.

Alethea Research recently released a report showing that 96% of those polled believed that the Internet will play a bigger role in the future of radio. Alethea also reported that of all the new revenue generating technologies that are currently available to the radio industry such as the additional programming channels offered by HD technology, the online streaming of a station's program content had the *highest* overall earning potential for generating brand new revenue.

Further supporting the importance of streaming content to the web is the latest report from BIA/Kelsey, which predicted radio industry advertising revenue increases for 2012 at around +3.5% for on-air and +15% for online. If station management hasn't yet taken a look at the online advertising aspect of streaming their content to the web, or worse yet, hasn't been or isn't taking it seriously, it's definitely time to give it a second look.

THE WORLD CAN HEAR YOU NOW

Given the huge potential audience existing well beyond a station's licensed coverage area it's obvious that even from a purely financial perspective a station should be streaming their program content to the web. For the best chance at snagging this new and potentially vast audience the station needs to be streaming at the best possible quality. Of course this leads to questions regarding which audio codecs and streaming processors to use, and I'll cover that in a moment.

WHERE DO THEY LISTEN?

An important piece of data revealed by a recent 'Radio Streaming News' survey was that most online listening is occurring between 8 AM and 6 PM. Such finding suggests that those surveyed were using their computers at work to listen to online streams. A work environment is usually a quiet place, so this listening is probably being done at low volume so as not to disturb nearby workers.

Being able to enjoy a web stream even at low volume requires consistent management of the stream's psychoacoustics. Competently designed audio processing can simultaneously reduce the codec's workload and better match the audio to a listener's local listening environment. Failure to take both challenges into account results in poorer codec performance (more coding artifacts) and cause a listener to continually adjust his volume control (annoying).

Remember that any listener able to listen to *your* stream is also able to listen to *other* streams. Said another way, your station's web stream is competing with a vast assortment of online entertainment, so its success as a business venture depends on at least the following criteria:

- It must have compelling program content;
- It must be easy to listen to;
- It must sound good.

Compelling programming is the responsibility of the station's programming department and is not a subject covered here. But if the ratings for a station's over-the-air signal are near the top, there's a good chance that its programming is compelling enough to please Internet stream listeners too.

A TYPICAL SETUP

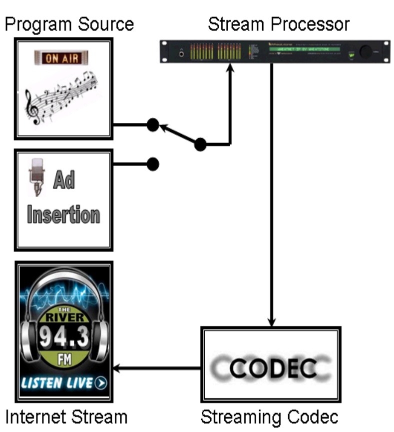


Figure 1

The signal chain for streaming can be quite simple. A close look at a typical chain reveals that in commercial applications there may be multiple program sources; one which carries (for instance) music intended for the local on-air signal *and* the Internet stream, and one that carries commercials or other content that is sold separately and is only for the stream. Regardless of the number of audio sources, the streaming processor must condition the audio before handing it off to the codec. The codec then turns the audio into a highly compressed data stream which is what stream listeners connect to in order to hear your station. The most important piece of equipment in the setup shown is the Stream Processor because it is the sole device responsible for conditioning the audio so the codec can create the best sounding Internet stream.

AND THE CODEC WINNER IS…

Forget MP3 – it's old school. The 'best codec?' question will stay open to debate because codec technology continues to evolve. At present however, Coding Technologies' AAC+ (or AACv2) is a popular choice for its ability to deliver remarkably good audio quality at reasonable bitrates. AAC can also be decoded by Adobe Flash Player 9 and up, making it compatible with popular browsers and operating systems. AAC is one of several audio formats currently supported by iPhone, iPod, Android, and mobile phones from Sony Ericsson, Nokia and others. The AAC codec has even begun to see adoption in factory-installed car audio too, though only in the 'luxury' car category so far.

A BETTER SOUNDING STREAM

A great sounding web stream gets that way by using dedicated and competent stream processing that's radically different from that used for 'on-air'. In fact I should mention this right now; 'on-air processing' is *entirely inappropriate* for processing audio for a web stream (*please* resist the temptation to feed your stream from an off-air tuner too!). Stations invest in special and expensive audio processing to make their over-the-air signal sound great. Web stream processing also has unique requirements, and using the correct processing for your web stream can help it be an overwhelming success.

PREPPING THE AUDIO

The first task that a stream processor must do is to manage the dynamic range of the program content and do it in ways that listeners don't notice. This task isn't trivial and involves several important steps that shouldn't be skipped. You must also ensure that audio levels into the processor remain within a certain range in order to minimize the influence of noise and distortion. This range is illustrated in Figure 2 where dashed lines represent the minimum and maximum audio levels that should be allowed.

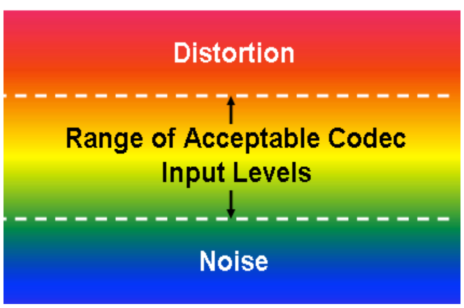


Figure 2

Next, the audio is processed to remove long-term level variations. This is accomplished by an intelligent level control algorithm specially designed to completely hide any hint to the listener that audio level modifications have occurred. Done with care and finesse, this step virtually eliminates any possibility that your web stream will be a source of listener fatigue.

Next, highly specialized algorithms operating in several frequency bands condition the audio to enhance the consistency of its loudness and intelligibility. This involves an automatic rebalancing of short-term levels *and* corrections to the medium-term spectral balance to create audio that is consistent and easy to listen to. As in the long-form level control previously covered, the processor should be able to perform these tasks on *any* kind of program material and without a listener ever noticing that something has been done. In fact, this is the processing stage that does most of the heavy lifting to ensure that a web stream remains listenable even at very low volume settings or in noisy environments.

The final two stages of the stream processor are the most sophisticated. Though the two stages *can* and *do* operate independently, on some program material they *can* and *do* behave as a unified algorithm. The nuances of this processing section are proprietary and critically important to the quality of the web stream because there are two mutually exclusive tasks at work; controlling peak audio levels to prevent overload within the codec, and predicting and mitigating certain kinds of audio content that would prove troublesome for the codec.

It is this final processing section that separates 'stream' processing from 'on-air' processing. It may seem that the 'front ends' of both processors are similar, and to an extent that is true. However, what may not be intuitive is that the two processor's 'back ends' are, and need to be, completely different in order to appropriately perform their tasks. Remember; the primary purpose of on-air processing is to *control modulation*. The primary purpose of stream processing is to *manage what the codec 'sees'*, audio-wise, as it turns your audio content into a drastically reduced number of digital bits that becomes your web stream.

THE 50,000 FOOT VIEW

I've explained how stream processing, or pre-codec processing as it is sometimes called, is a completely different animal than on-air processing, and also why each would perform poorly in the other's application. The importance of this difference cannot be overstated.

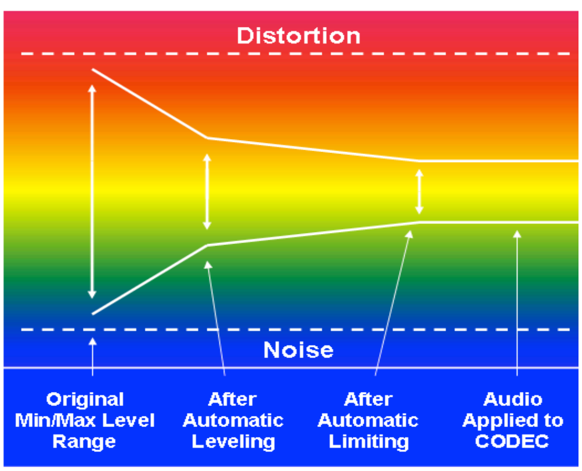


Figure 3

In Figure 3, at left, is a simplified view of how the audio levels are managed inside the stream processor. Although other processing functions or activities have been alluded to, only the management of audio *levels* has been described in any detail thus far. Audio consistency and minimal coding artifacts will help to make listeners comfortable listening to your web stream and may even enticethem to listen longer. Now it's time to dig deeper…

EVERY CODEC NEEDS A ONE CALORIE DESSERT

The *only* purpose of an audio codec is to take a whole lot of audio data and throw away things we probably couldn't hear anyway and keep doing it until there's only a little bit of the original data left. With the codec's task now understood, one way we can help it do a better job is to never hand it things to code that either aren't relevant to our audio, or might be difficult for it to turn into good sounding data. Even though the codec discards *most* of the original audio data, things should still sound pretty good if our expectations remain reasonable and we don't do anything to make its job even harder than it already is. That said my own "Top Five" for making sure that a codec is ingesting only the things it should be are:

- Don't try to send 'Hi-Fi' content to 'Lo-Fi' listening environments;
- Avoid sending the codec sustained noises such as hiss and hum;
- Minimize any errors in left/right channel balance;
- Avoid distortions due to badly clipped audio;
- Avoid overly boosted high frequencies.

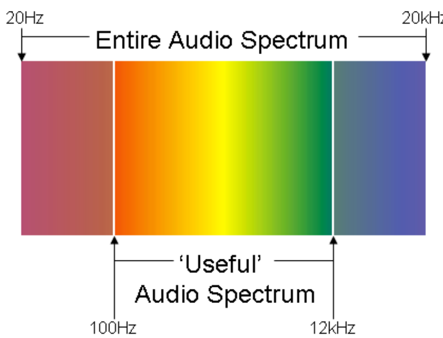


Figure 4

Figure 4 at left hints that audio frequencies beyond the reproduction range of a listener's equipment can consume codec resources better reserved for coding audio that *everyone* can hear. Several informal surveys that we performed revealed that most web stream listening is done on equipment with 'Lo-Fi' specifications. Does it make sense then to hand the codec audio content that can't even be heard on the majority of your stream's listener's systems? In a word: No.

Removing audio content that can't be heard on the systems listeners use to hear your stream can dramatically improve its subjective audio quality. As Figure 4 shows, audio frequencies below about 100 Hz and above about 12 kHz can usually be reduced or even completely removed with no negative impact on the *perceived* quality of your stream. In fact, doing so can actually help your stream sound *better* due to reduced or removed 'codec teasers' such as hiss and hum.

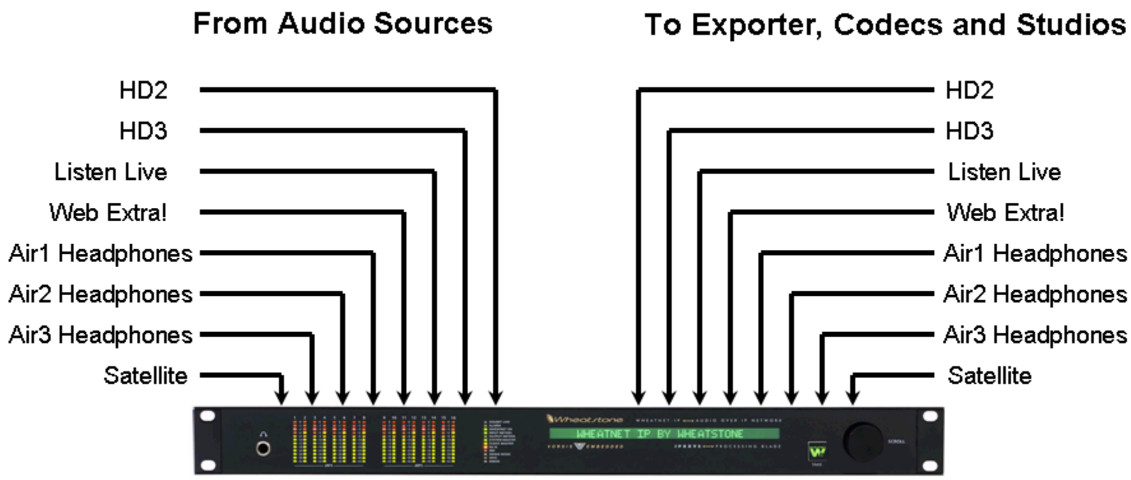
The lower the codec's bitrate the fewer bits there are for coding your station's audio. Seemingly unimportant issues such as left/right channel balance can become critical at low codec bitrates. Further, below about 64kb/s I recommend using mono operation because although the stream may no longer be 'in stereo' it will certainly have a much higher *apparent* quality to the stream's listeners.

Beyond choosing the codec to use you must also choose a stream processor capable of preconditioning your content so that it always sounds great to your stream's listeners. One product that stands tall above all others in the stream processing category is Wheatstone's Aura8ip 'Processing Blade' with Vorsis Embedded audio processing. The Aura8ip was *born* for streaming.

Wheatstone Aura8ip – 'Processing Blade'

Onboard Aura8ip are *eight* independent multiband Vorsis streaming processors equipped with the ability to use three different audio I/O technologies; Wheatnet-IP, AES3 digital, and analog. It can even operate as a completely standalone product where needed.

One growing trend is transmitting 'extra' audio streams *other than* the main over-the-air analog and HD programming to the Internet (shown as Web Extra! below). These are known as 'side channels' and could be special programs and a new source of revenue. Aura88ip can provide processing for all of the station's HD channels, their Internet streams including 'side channels', and still have processing channels left over for signals such as talent headphone feeds. The graphic below illustrates how many audio sources can be processed with just *one*



Aura88ip Stream Processor

AurA88ip.
The cost effective Aura8ip Audio Processing Blade marries two of Wheatstone's core technologies (Vorsis Audio Processing and WheatNet-IP Intelligent Network). It is an efficient way to provide audio processing anywhere within a facility.

Below is a list just some of Aura8ip processing features:

- High pass filter to remove undesired subsonic energy and hum;
- Two band parametric equalizer with LF and HF shelving equalizers;
- Phase linear 1, 2, or 3 band AGC;
- Intelligent multiband AGC dynamics control + stereo enhancement;
- Peak responding multiband limiter and bass enhance;
- Specialized Codec Conditioning + Final Lookahead Limiter.
- Available with 8 digital, 8 analog or a 50/50 mix of analog and digital I/O.

Included with Aura8ip is Vorsis' acclaimed "Audio Processing Guru®" software which allows easy setup and tuning of the processing using familiar, straightforward controls. We also make available our most sophisticated control interface called "GUI Pro," which provides access to *every* processing parameter allowing for expert-level sound customization.

Because it's a BLADE, Aura8ip can also instantly configure itself as part of a new or existing WheatNet-IP Intelligent Network and make its processing power available throughout that network. The Aura8ip is easily configured and controlled over a standard Ethernet network using a laptop or desktop computer. When part of a Wheatnet-IP audio network Aura88ip permits 8 channels of audio I/O, control, metering, and even processor tuning, all on one Cat5e Ethernet cable!

One thing that may not be so obvious about the Aura88ip's feature set is that because it is a Wheatnet-IP Blade, it is *also* an audio mixer *and* an audio router. This means that Aura88ip can do the switching and mixing of multiple audio input sources to any of its eight processing instances, and also the routing of the outputs of those processing instances to *any* output on its back panel or to *any* destination in an entire Wheatnet=IP system! And can do it hands off!

The Aura8ip's embedded Vorsis Processing makes your stream's programming sound its very best – regardless of where your stream is coming *from* – and regardless of where your listeners *are*. With both Wheatstone and Vorsis design pedigrees on board, Aura8ip was *born* for streaming!