



**Writer**  
Rick Bidlack  
Development Engineer,  
Wheatstone Corp.

## A primer on the politics & protocols of streaming

What to know about getting your streams cloud-ready and CDN-compatible

*This article is based on a paper prepared for the 75th NAB Broadcast Engineering and IT Conference. Presentation videos and proceedings are available at <https://nabpilot.org/beitc-proceedings/>.*

**T**he broadcast industry is in a transitional phase, heading toward a world in which a large portion, perhaps even a majority, of listeners will receive the station's program via an internet stream rather than terrestrial radio broadcast.

Many accomplished engineers and station managers, while experts in traditional radio technology, find themselves feeling like neophytes once again when it comes to internet streaming. This paper aspires to present an instructional overview of what's involved in this generation of an internet audio stream, as well as a better understanding of where standards and practices are still developing.

### **Signal flows**

Streamed content starts out on the same path as broadcast content, typically originating from the automation system if it's music or from the studio microphone if it's voice, or both.

One, however, goes to the broadcast chain and the other goes to the streaming encoder, and that one difference introduces new considerations in both processing and metadata for streaming. To understand these considerations, it helps to get a closer look at the signal flow of each.

### **Broadcast signal flow**

For broadcasting purposes, content is fed into an automation system, which sends its audio output to an FM (or AM) processor, where it is processed and multiplexed, and then passed on to a transmitter that disseminates the program to the public.

As a side channel, the automation system has also been stocked with metadata (artist, title, album, duration, label, ISRC, etc.) for every piece of audio in its playlist; this metadata is also transmitted to the FM processor, where it is turned into an RDS signal that rides as a sideband on the MPX output.

There are, of course, variations in this flow path. For example, the metadata from the automation machine may be routed instead to an “aggregator” that cleans and ensures the accuracy and consistency of the metadata before it is forwarded to the FM processor. Or it might not even go to the processor, but to a separate RDS encoder, the output of which is then mixed with the MPX from the processor.

The exact details may change, but the steps along the path remain essentially the same.

## Stream signal flow

For streaming purposes, content enters the automation system as before, but in this case the output is routed to a stream encoder (also called an “origin server,” for reasons that will be clear later), and from there to a CDN, or Content Distribution Network.

The stream encoder has three jobs:

1. Process and condition the audio signals, optimizing it for the compression algorithms;
2. Encode, packetize and transmit the program over the public internet to the destination server (the CDN);
3. Handle the reformatting and forwarding of metadata from the automation system on to the CDN.

## CDN streaming and add-ons

Before looking at the inner workings of the stream encoder, let’s examine the Content Distribution Network. The CDN

“ **The stream encoder is also a mediator/translator between the automation and CDN, transforming the format of what it ingests into one that the CDN requires.** ”

plays the role of the transmitter in the streaming paradigm, but its unique position in the program flow path actually gives it a bigger and more important job.

The main function of the CDN is to serve your stream to thousands or tens of thousands of listeners. But the fact that your program and all associated metadata pass through the CDN’s servers, and they know who is listening from what location and for how long, gives them an opportunity to provide a whole suite of add-on services.

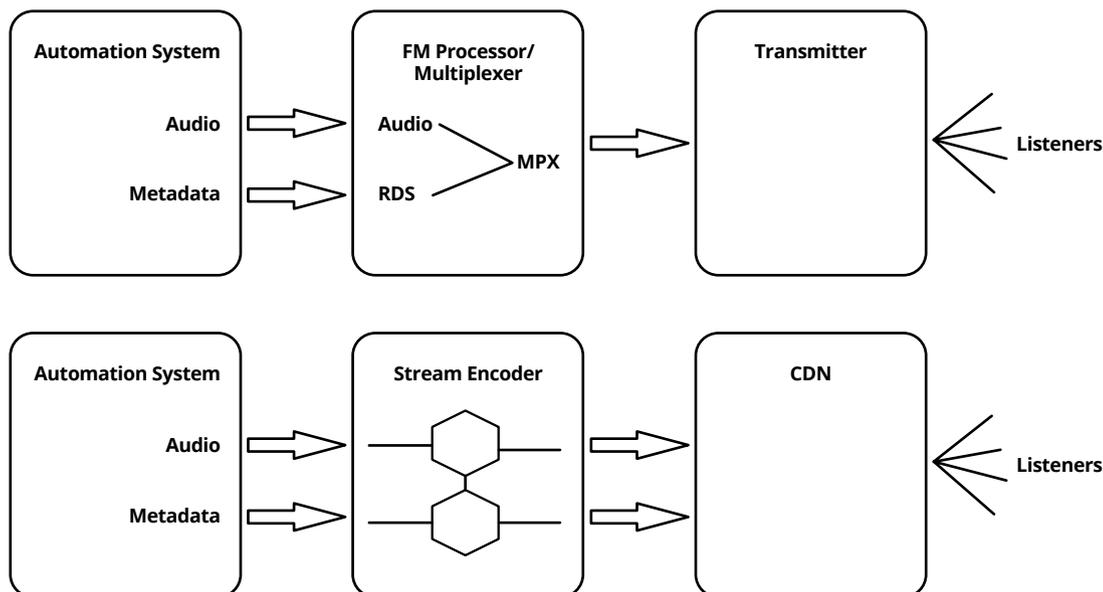
A big one is ad replacement, which is usually geographically based but could also be tailored to whatever can be deduced about the individual listener’s tastes and habits.

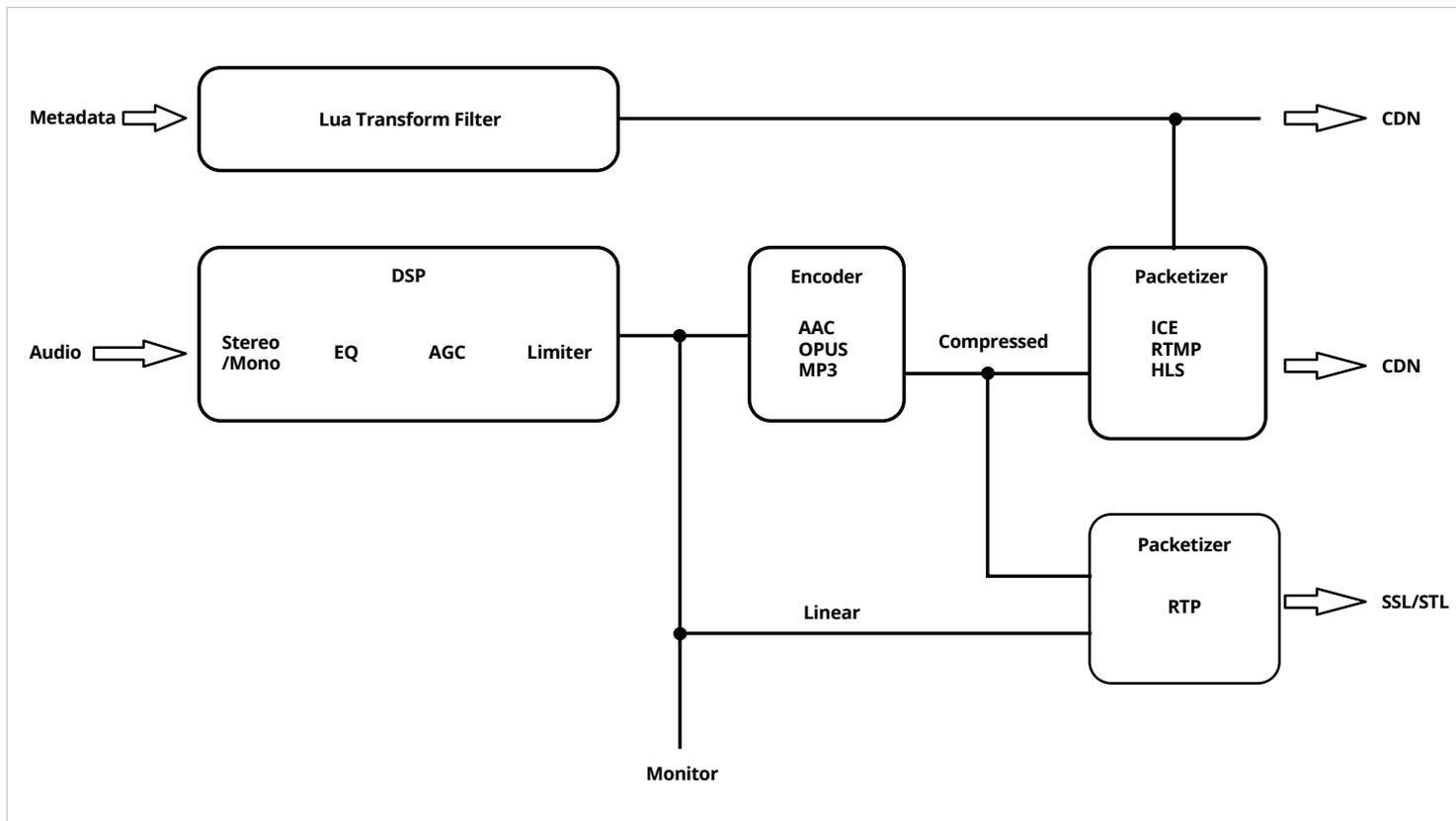
Geo-blocking, logging, skimming, catch-up recording and playback, access to additional metadata (e.g. album art, fan club URLs), listener statistics and click-throughs, customized players, royalty tracking, redundant stream failover, transcoding from one format to another — these are some of the services that CDNs typically provide.

Thus, the CDN basically controls the distribution of the stream to the listening public. It is the responsibility of the stream encoder — the origin server to the CDN’s ingest and distribution servers — to make sure that the CDN gets the right data at the right time and in the right format.

Especially with regard to metadata, the CDN determines what the format will be. The stream encoder is therefore also a mediator/translator between the automation system and the CDN, as it must be able to transform the format of whatever it ingests into the format that the CDN

**Below**  
Typical signal flows in broadcast, top, and in streaming.





**Above**  
The encoder's central function is data compression.

requires. With that in mind, let's take a closer look at the stream encoder.

### Job 1: Data compression

The stream encoder performs various functions. The central function, which all others are designed in relation to, is to apply data compression to the audio stream, thereby reducing the bandwidth required both to transmit the stream up to the CDN, and for the listener to receive it.

By far, the most widely used compression codec for high-quality audio storage and transmission is AAC, but its predecessor MP3 is still used for legacy streams (to support legacy players). All compression algorithms operate by reducing the information contained in the audio signal to as few bits as possible; you therefore want to maximize the

value of those bits by removing extraneous artifacts from the audio signal before it hits the compression codec. That cleansing and conditioning of the raw audio signal is the job of the DSP section at the input.

### Encoder handoff to CDN

The compression codecs produce periodic output in the form of chunks of data, each of which represents a small segment (typically 5 to 20 milliseconds) of the continuous audio signal. These chunks are wrapped in a transport format and transmitted to the CDN's ingest server as discrete, time-stamped packets. Common transport protocols include Icecast, RTMP, HLS, MPEG-DASH and RTP.

Metadata is typically received by the stream encoder on a TCP or UDP socket, and most commonly arrives formatted as XML. What happens after that depends on the transport protocol being used.

For Icecast streams, metadata updates (including ad triggers) are sent to the server out-of-band, as separate HTTP messages. For RTMP, the metadata update is embedded in the transport stream itself, as a special INFO type packet. For HLS, metadata may be reformatted as ID3 data and embedded into a separate channel of the underlying MPEG2 Transport stream, and ad triggers can be woven into the manifest file as SCTE-35 "program replacement opportunities."

“Even in the age of digital streaming, keeping entropy at bay is still the constant battle.”



## Read the Proceedings

Radio World thanks the organizers of the NAB Broadcast Engineering and IT Conference for their cooperation in bringing this and other event content to our pages. We encourage readers to explore the available resources at <https://nabpilot.org/beitc-proceedings/>.

For all of these methods, the exact details can differ from one CDN to the next since there are no universally accepted standards for handling metadata, so setting up a stream and getting the metadata to update correctly and in sync with the audio stream often requires a short period of trial and error while negotiating protocols with the CDN.

RTP is a special case. RTP is the only stream format that is commonly used to carry uncompressed, full-bandwidth audio (in the form of AES67, for example), although it can be used to carry virtually any kind of data, including AAC, Opus or MP3 audio.

RTP is typically used in a studio-to-studio or studio-to-transmitter link, rather than as an ingest feed to a CDN.

## Metadata formatting

The final area we want to examine is the metadata itself. We've been talking about transforming it from one format to another, but what does that mean, and why do we need to do it?

To answer, let's look at some examples of raw metadata received from various automation systems. (Names and numbers have been changed to protect the innocent.)

Some systems export simple tagged text:

```
artist=Sonny Rollins
title=God Bless the Child
length=00:07:29
```

More commonly, metadata arrives in XML or XML-like form. Note that many fields may be empty:

```
<nowplaying><sched_time>220200</sched_time><air_time>402000</air_time><stack_pos></stack_pos><title>TheSkylsANeighborhood</title><artist>FooFighters</artist><trivia>*</trivia><category>MNJ</category><cart>R585</cart><intro>14000</intro><end></end><station>93.5HD1</station><duration>243300</duration><media_type>SONG</media_type><milliseconds_left></milliseconds_left><Album>ConcreteandGold</Album><Field2></Field2><ISRC>USRW9170028</ISRC><Label>RCA</Label><Tempo></Tempo><Year>2017</Year></nowplaying>
```

The HTML ampersand entity in this one will require special handling:

```
<audio ID="id_3155460704_30756200">
<type>Song</type>
<status>Playing</status>
<title>JINGLE BELL ROCK</title>
<artist>Hall & Oates</artist>
<length>00:02:03</length>
<category>Classic</category>
</audio>
```

Here's a station ID and a liner from the same source. Some CDNs may want this data, some may not:

```
<audio>
<type>Link</type>
<status>Playing</status>
<artist></artist>
<title>ROCK91.7 | LEGAL ID2</title>
<number>120002</number>
<length>00:00:11</length>
</audio>
```

```
<audio>
<type>VoiceTrack</type>
<status>Playing</status>
<artist></artist>
<title>VT TLA3 - Rock: 2021-03-25 08:37</title>
<number></number>
<length>00:00:06</length>
</audio>
```

Here's a sweeper and an ad from another source, below. Notice that the `<media_type>` tag distinguishes one from the other. The arrival of a SPOT is often used to trigger an ad replacement by the CDN. Durations shown here are in milliseconds.

```
<nowplaying><sched_time>67199000</sched_time><air_time>67267000</air_time><stack_pos></stack_pos><title>CYQTONEQUICKSWEEP#12DRY</title><artist>201971108:06:28</artist><trivia></trivia><category>IM5</category><cart>BIBE</cart><intro>0</intro><end></end><station>95.7KQED</station><duration>3900</duration><media_type>UNSPECIFIED</media_type><milliseconds_left></milliseconds_left><Album></Album><Field2></Field2><ISRC></ISRC><Label></Label><Tempo></Tempo><Year></Year></nowplaying><nowplaying><sched_time>33716000</sched_time><air_time>33721000</air_time><stack_pos></stack_pos><title>ShopLocalAWestfieldMerchantsFriday</title><artist>WestfieldMerchantsJune</artist><trivia>shoplocalgreencountry</trivia><category>COM</category><cart>4652</cart><intro>0</intro><end></end><station>95.7KQED</station><duration>30000</duration><media_type>SPOT</media_type><milliseconds_left></milliseconds_left><Album></Album><Field2></Field2><ISRC></ISRC><Label></Label><Tempo></Tempo><Year></Year></nowplaying>
```

None of the metadata messages in these examples can be transmitted to the CDN ingest server in the form they are in. Instead, the relevant data must be stripped out and reformatted into a protocol that the ingest server can

