

PETER
BLATCHFORD

STARFISH TECHNOLOGIES

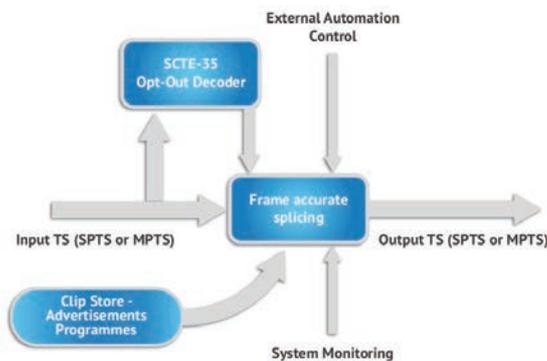


MAKE THE SWITCH

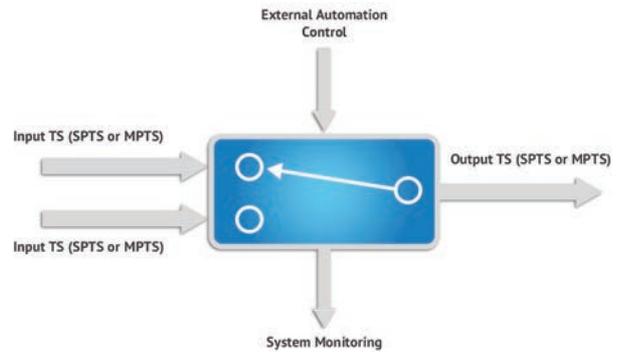
Hardware-based ad servers and switching solutions have been around for a long time and, broadly speaking, do the job just fine.

That said, after a number of meetings before and during NAB 2017, what I found is that a lot of people are looking to replace components of their hardware-based ad servers, the switching function being chief among them. There's nothing particularly wrong with existing ad servers themselves. They've been installed and working for a long time. But therein, perhaps, lies the issue. While the ad servers can be operating OK, the switching function is not as efficient, or elegant, for switching modern-era transport streams.

This is partially what led Starfish to develop a software-based splicer, TS Splicer that happens to include a software-based switch. The feedback we have had from the marketplace is that while TS Splicer has lots of useful applications (always nice to hear), there is also a need for a straightforward, software-based switch, which is how we came to develop TS Switch.



The US market in particular is very mature in terms of ad insertion. Many broadcast facilities have had an ad server as part of their infrastructure for years, so when some of them said, "TS Splicer is a great product, but we may want to leave our existing ad insertion infrastructure in place, and just replace the switch technology", for now anyway. That's why we developed a new way to switch that runs on generic 'off the shelf' hardware. It's scalable and flexible, but most important, it provides a very technically clean switch.



It's also ideal for redundancy applications because the switch has an auto-failover capability. When you have a main and standby server playing out an important channel, you need an automated switch downstream in case one of the streams fail. If that happens, the signal automatically detects an input failure and instantly fails over to the alternate path. In the case of TS Switch, it also has a mode for identical, co-timed streams, which basically involves buffering. What that means is that the switches, when they occur, are transparent visually, and within the structure of the stream.

The software-only approach to switching benefits broadcasters and distributors because some of their hardware can stay right where it is. There's no great disruption or expense to replace all the existing infrastructure. And because the switch software is scalable, and flexible in terms of the bit rate of the media and the media encoding format, the user can switch up to 20+ channels on a single server.

Switching SDI signals is straightforward, but switching encoded streams – particularly when you have two, disparate, signals - and doing it well, is incredibly hard. Because the GOP boundary between two signals will not line up in encoded stream, you have to be a lot smarter about the way you handle the switch.

The first goal is to produce an output stream that maintains the structure of the transport stream. What that means is that anything downstream, such as decoders and any other ancillary equipment, will not be thrown off course by the switching process. In fact, they will not see or sense any discontinuity and continue operating as normal.

We have developed a product that is software based, scalable, switches transport streams cleanly and is flexible in terms of bit rates and encoding formats.

It may be time for you to make a switch. ■