SCTE•ISBE
CABLE-TEC EXPO®
VIRTUAL EXPERIENCE » OCTOBER 12-15 2020

2020 Fall
Technical Forum
SCTE·ISBE • NCTA • CABLELABS®

Satellite to Fiber Broadcast Execution With SCTE 35 and 224

Implementing Linear Rights and Alternate Content for Network and Affiliate Feeds

A Technical Paper prepared for SCTE•ISBE by

Stuart Kurkowski, PhD
Distinguished Engineer, Principal Architect
Comcast Technology Solutions
Dry Creek Facility
303-503-2680
Stuart_Kurkowski@comcast.com

Neill Kipp
C&SP Architect Lead
Comcast Technology Solutions
1899 Wynkoop Street, Suite 500
720-530-6917
Neill_Kipp@comcast.com

Table of Contents

Title **Page Number**

List of Figures

Title **Page Number**

List of Tables

1. Introduction

The world of broadcast video delivery is evolving rapidly, especially with the proceeding by the Federal Communications Commission (FCC) to clear the lower 280 MHz of the C-band spectrum, to pave the way for its use by 5G services (FCC Report and Order). Satellite companies including Intelsat and SES use the C-band spectrum to serve TV broadcasters and cable network operators with video feeds. The FCC ordered these companies to move their operations to the upper 200 MHz of C-band in order to prevent interference from mobile services (Reardon 2020).

As a result of the uncertainty of C-band going forward and the continual demands for higher video quality, broadcasters and content providers are considering transitioning more fully from satellite to fiber, and/or Internet-based delivery mechanisms. Linear channel delivery over terrestrial optical fiber offers lower cost, lower latency, lower bandwidth constraints, and higher quality than a corresponding delivery over satellite. With these benefits and the FCC mandates, operators have some compelling reasons to make the transition from satellite to fiber.

Any transition to terrestrial video delivery over fiber must occur with minimal impact to current customer workflows. The risks of transition certainly include disruption of primary service, but also of signaling for advertising, local contributions, and alternate programming, including blackouts. The benefits include additional over-the-top (OTT) distribution. Certain hybrid use cases increase disruption risk. For example, national feeds arriving by satellite can signal a local contribution arriving over fiber, or vice versa.

The SCTE Digital Video Subcommittee (DVS) maintains a family of standards that, when implemented together, help support a seamless transition from satellite to fiber.

Regardless of transmission path, linear streams carry in-band signaling following ANSI/SCTE 35, the “Digital Program Insertion Cueing Message for Cable” standard. ANSI/SCTE-35 signals convey the start and end of advertising, programming boundaries, and license boundaries.

The “Event Scheduling and Notification Interface” (ESNI; ANSI/SCTE-224) is an out-of-band signaling standard that allows the enforcement of viewing restrictions, such as web embargos, startover, lookback, and alternate programming. ANSI/SCTE-224 provides the mechanism to achieve the video markup and execution for all aspects of the workflow, including national and local contributions, ad insertion, as well as execution on the playout side for local broadcasts and OTT content.

And, the “Real-time Event Signaling and Management API” (ESAM; ANSI/SCTE-250) provides a software standard for converting in-band ANSI/SCTE-35 or out-of-band ANSI/SCTE-224 signals into real-time requests of a signal decision system. ANSI/SCTE-250 can be used to implement advertising placements, license constraints, and local programming.

A linear signal processing system that utilizes the combination of ANSI/SCTE-35, 224, and 250 requires a responsive and scalable architecture that can be implemented using a combination of on-premise systems and cloud-based services (Figure 1). Implementing this linear system enables the seamless transition of fully featured linear channels from satellite to the Internet.

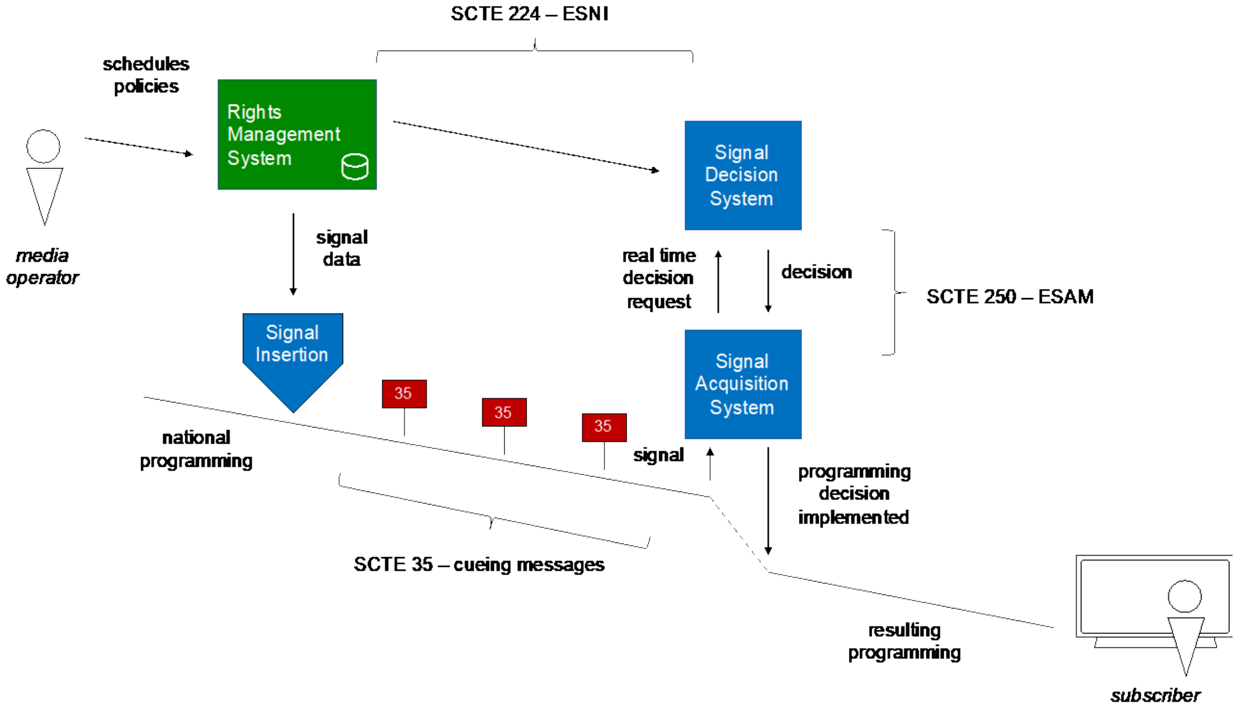


Figure 1 - High Level Architecture

2. Standards Background

The interplay between content rights and multi-platform distribution has become increasingly complex. Every component in a delivery workflow must weigh simplicity and deployment agility against quality of experience. Today, configuration data is manually entered into spreadsheets and application programming interfaces (APIs) that are distributed via multiple channels. Programmers are reliant on operators to acknowledge that data has been received, and then operators must ingest data from multiple sources and formats before it can be assembled and delivered. By utilizing standards created by the SCTE DVS and specifically Work Group 5, Digital Program Insertion, all parts of the workflow can work together at a machine-to-machine level. In the following sections we detail these various standards the enable this machine-to-machine workflow.

2.1. ANSI/SCTE 224

ANSI/SCTE 224 ESNI provides a robust framework of Extensible Markup Language (XML) messages. It details descriptions of audience characteristics, and viewing policies associated with each audience, it also allows for channels (Media) and individual events (MediaPoints) to describe start and end times (MatchTime) or in-band signaling (MatchSignal) information. With ANSI/SCTE 224, operators can implement bindings from events and audiences to specific programming signals; this will signal downstream systems the desired programming changes and exactly when to implement them. Although the framework is robust, operators must assemble meticulous programming details into a standard format, control metadata delivery targets, maintain metadata visibility across a diverse delivery ecosystem, and reliably communicating the resulting instructions to every relevant distribution point.

ANSI/SCTE 224 allows for rights convenience by setting up five different message types to carry the information: `Media, MediaPoints, Policy, ViewingPolicy`, and `Audience`. The following sections detail each of these message types and provide a brief description of each.

2.1.1. Media

The `Media` object represents a linear television channel and its schedule of events. The `Media` is the container that contains the `MediaPoints` used to describe individual events that are scheduled to occur on that channel.

2.1.2. MediaPoints

`MediaPoints` are individual events that occur within a channel or a show. These events are traditionally program start, ad start, ad stop, and program end. The `MediaPoint` is matched against an event in the video either by time-based triggers (e.g., program start at 2 PM) or matching against the SCTE 35 in-band signal described below. Using the in-band signal has the advantage of allowing the actions to be taken on a frame-accurate basis, making for a better user experience. When a `MediaPoint` is matched to an in-video event, it is "triggered," which means that one or multiple `Policy` messages can be applied to the state of the channel. The `MediaPoint` can also contain additional metadata such as alternate identifiers and electronic program guide (EPG) information. This enables live to video on demand capture or accurate EPG usage with the out-of-band ANSI/SCTE 224 data. See Appendix A for a complete example of a ANSI/SCTE 224 `Media` object.

2.1.1. Policy

The `Policy` message is a container for `ViewingPolicy` that can be associated with an event (Figure 2). `Policy` elements are added and removed to control channel playout. For example, a `Policy` about trick mode restrictions can be added to the channel state and a second `Policy` that sends an audience to an alternate channel can be added as well. It is also at the `Policy` level that items are removed from the state, so in the previous example, if the `Policy` to send the audience to an alternate channel is removed from the state, then that audience will return to watching the main channel.

```
<Policy xmlns="http://www.scte.org/schemas/224"
  id="xyz/policy/any/slate"
  lastUpdated="2016-11-18T15:00:00.000Z">
  <ViewingPolicy xmlns="http://www.scte.org/schemas/224"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xlink:href="xyz/viewingpolicy/any/slate">
  </ViewingPolicy>
</Policy>
```

Figure 2 - Example ANSI/SCTE 224 Policy Object

2.1.1. ViewingPolicy

`ViewingPolicy` elements are key messages, because they bring together the action for a triggered event and the appropriate audience. So, for example, the `ViewingPolicy` for one audience might be to message telling the view the programming is unavailable, while a different audience might be sent to an alternate feed (Figure 3). Defined sets of actions cover content directives, trick mode restrictions, dynamic ad insertion, capture controls for recording, and signal insertion and deletion.

```
<ViewingPolicy xmlns="http://www.scte.org/schemas/224"
  id="xyz/viewingpolicy/any/slate"
  lastUpdated="2016-11-18T15:00:00.000Z">
  <Audience xmlns="http://www.scte.org/schemas/224"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xlink:href="xyz/audience/location/any">
  </Audience>
  <Content xmlns="urn:scte:224:action">urn:scte:224:action:slate</Content>
</ViewingPolicy>\
```

Figure 3 - Example ANSI/SCTE 224 ViewingPolicy Object

2.1.1. Audience

`Audience` is individual messages used to characterize the groups of viewers impacted by a `ViewingPolicy` (Figure 4). Numerous standard audience characterization elements include ZIP codes, latitude and longitude, device types, and viewer status.

```
<Audience xmlns="http://www.scte.org/schemas/224"
  id="xyz/audience/location/any"
  description="No Travel Restrictions"
  lastUpdated="2016-10-22T16:00:00.000Z"
  match="NONE">
  <Zip xmlns="urn:scte:224:audience">00000</Zip>
</Audience>
```

Figure 4 - Example ANSI/SCTE 224 Audience Object

Combining these five message types into sets of machine-to-machine executable instructions is the key to utilizing ANSI/SCTE 224 to manage the execution of content, at scale, across multiple distribution partners.

2.2. ANSI/SCTE 35

ANSI/SCTE-35 in-band signals appear in MPEG-2 video transport streams and narrate events as they play out. ANSI/SCTE-35 signals primarily mark when programs start or stop and when ad insertion should start and stop. Signals can also indicate emergency action messages. Each signal has an eight-bit segmentation_type_id that tells the receiver what type of event is being signaled (0x00 - 0xFF).

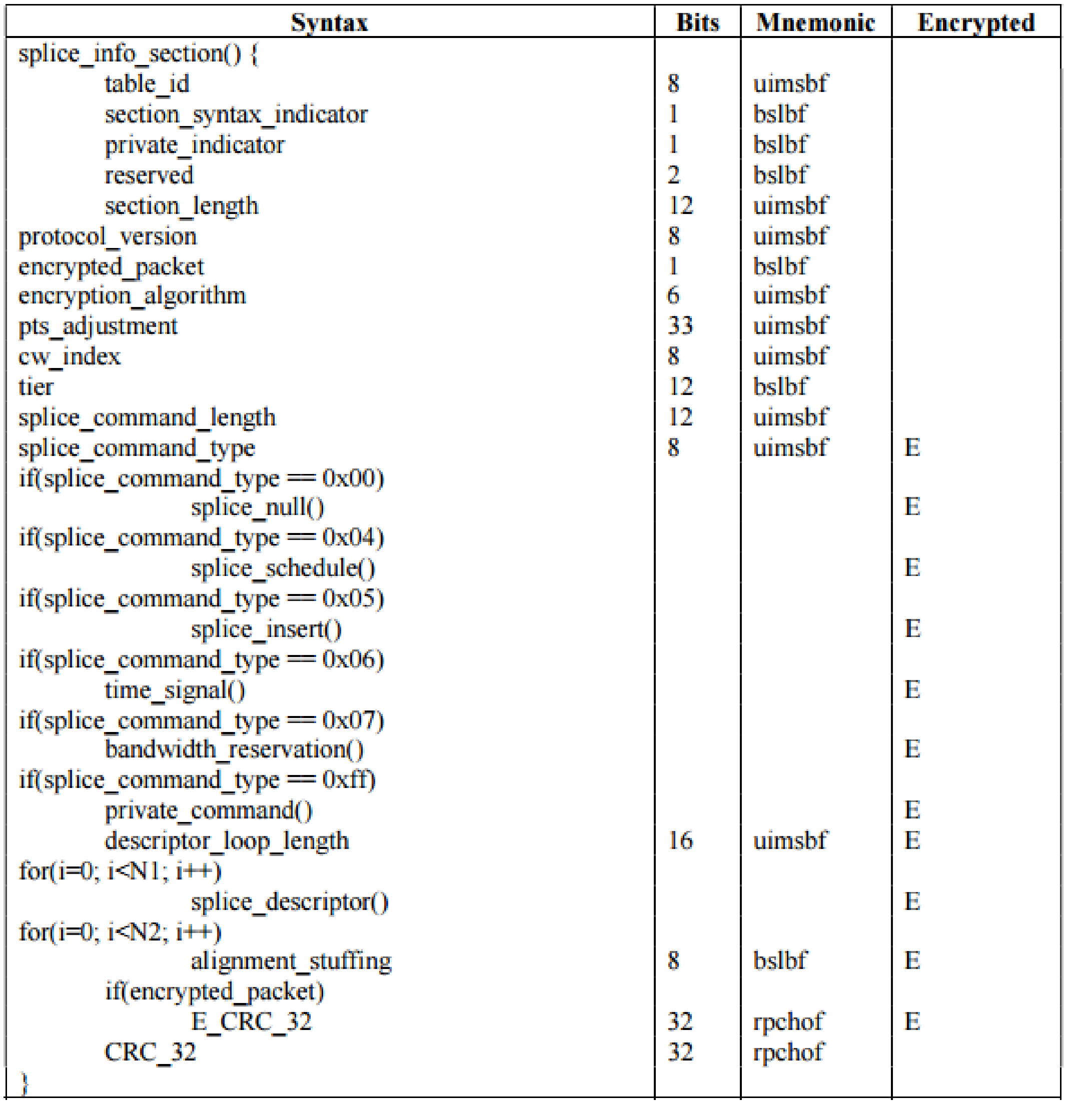
For example, if a baseball game started at 1 PM and ended at 3:45 PM, inside the stream would be a 0x10 signal at 1 PM for the program start and a 0x11 signal at 3:45 PM for the program end (and then probably a 0x10 signal at the same time for the next program start). There would be many 0x34/0x35 pairs that indicate when advertising placement opportunities appear. Program boundary signals 0x10 and 0x11 contain program identifiers, also known as unique program identifiers (UPID), that bind to ANSI/SCTE-224 rules based on programs.

In other words, programmers notify downstream devices by putting ANSI/SCTE-35 messages in a stream to signal ANSI/SCTE-224 channel events.

2.2.1. ANSI/SCTE-35 Serialization

ANSI/SCTE-35 has two basic serialization forms. The most basic form is a raw binary sequence. Table 1 shows an example. What’s shown is just part of the SCTE-35 message, not the whole container.

Table 1 - Example ANSI/SCTE 35 UPID Types



Syntax	Bits	Mnemonic	Encrypted
splice_info_section() {			
table_id	8	uimsbf	
section_syntax_indicator	1	bslbf	
private_indicator	1	bslbf	
reserved	2	bslbf	
section_length	12	uimsbf	
protocol_version	8	uimsbf	
encrypted_packet	1	bslbf	
encryption_algorithm	6	uimsbf	
pts_adjustment	33	uimsbf	
cw_index	8	uimsbf	
tier	12	bslbf	
splice_command_length	12	uimsbf	
splice_command_type	8	uimsbf	E
if(splice_command_type == 0x00)			
splice_null()			E
if(splice_command_type == 0x04)			
splice_schedule()			E
if(splice_command_type == 0x05)			
splice_insert()			E
if(splice_command_type == 0x06)			
time_signal()			E
if(splice_command_type == 0x07)			
bandwidth_reservation()			E
if(splice_command_type == 0xff)			
private_command()			E
descriptor_loop_length	16	uimsbf	E
for(i=0; i<N1; i++)			
splice_descriptor()			E
for(i=0; i<N2; i++)			
alignment_stuffing	8	bslbf	E
if(encrypted_packet)			
E_CRC_32	32	rpchof	E
CRC_32	32	rpchof	
}			

This signal can also be expressed in hexadecimal, as in a #EXT-X-DATERANGE tag in M3U8 (Figure 5).

```
#EXT-X-DATERANGE:ID="splice-6FFFFFF0",START-DATE="2014-03-05T11:15:00Z",PLANNED-
DURATION=59.993,SCTE35-
OUT=0xfc304c000000000000fffff00506fee36333f70036021943554549 00882da97f80010a3030303839
32343538351000000219435545490 07ef2567f80010a3030303833313935373411000058b3f887
```

Figure 5 - Example ANSI/SCTE 35 Hexadecimal Tag

It can also be expressed as Base64, as in ESAM messages, as described in section 2.3 (Figure 6).

```
<signaling:Binary
signalType="SCTE35">/DBMAAAAAAAA///wBQb+42Mz9wA2AhlDVUVJAIgtqX+AAQowMDA4OTI0NTg1EAAAAh
lDVUVJAH7yVn+AAQowMDA4MzE5NTc0EQAAWLP4hw==</signaling:Binary>
```

Figure 6 - Example ANSI/SCTE 35 Base64 Tag

The M3U8 EXT-X-SCTE35 and EXT-OATCLS-SCTE35 tags also use a Base64 implementation (Figure 7).

```
#EXT-X-
SCTE35:TYPE=0x10,TIME=1450707450000,ELAPSED=0,ID="dAQpTUaQSjOti/JZTqECfQ==",CUE="/DBMA
AAAAAAA///wBQb+42Mz9wA2AhlDVUVJAIgtqX+AAQowMDA4OTI0NTg1EAAAAhlDVUVJAH7yVn+AAQowMDA4MzE
5NTc0EQAAWLP4hw=="
...
#EXT-OATCLS-SCTE35:/DAlAAAAAAAAAP/wFAXwAAACf+/+AM3Qkf4AFJlwAAEBAQAAGkXGSg==
```

Figure 7 - Example ANSI/SCTE 35 M3U8 Base64 Tag

2.2.2. *XML*

The ANSI/SCTE 35 standard describes an XML serialization of ANSI//SCTE 35 as shown in Figure 8 (with namespace declarations omitted for readability).

```
<SpliceInfoSection protocolVersion="0" ptsAdjustment="0" tier="4095"
xmlns="http://www.scte.org/schemas/35/2013a"
xmlns:ns2="http://www.thistech.com/schemas/scte35/1">
        <EncryptedPacket encryptionAlgorithm="0" cwIndex="255"/>
        <TimeSignal>
                <SpliceTime ptsTime="4151498432"/>
        </TimeSignal>
        <SegmentationDescriptor segmentationEventId="1"
segmentationEventCancelIndicator="false" segmentationTypeId="17" segmentNum="0"
segmentsExpected="0">
                <DeliveryRestrictions webDeliveryAllowedFlag="false"
noRegionalBlackoutFlag="false" archiveAllowedFlag="false" deviceRestrictions="0"/>
                <SegmentationUpid
segmentationUpidType="1">30303039353532303839</SegmentationUpid>
        </SegmentationDescriptor>
</SpliceInfoSection>
```

Figure 8 - Example ANSI/SCTE 35 XML Serialization

2.3. ANSI/SCTE 250

The “Real-time Event Signaling and Management API” (ESAM; ANSI/SCTE-250) protocol is sufficient for communication between a signal acquisition system (SAS) and signal decision system (SDS). It describes the format and content of the messages required by the SAS to obtain instructions from the decisioning service for handling in-band ANSI/SCTE 35 signals (Figure 9).

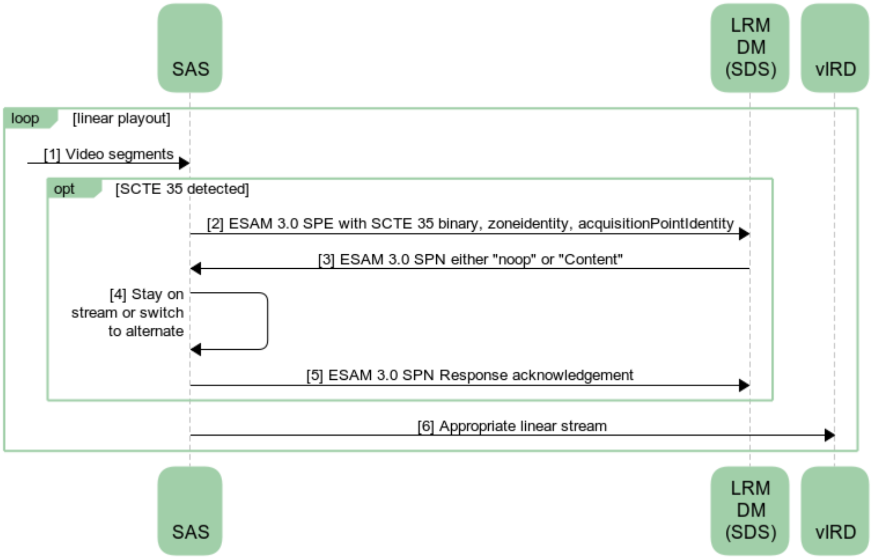

SAS
LRM DM (SDS)
vIRD
loop [linear playout]
[1] Video segments
opt [SCTE 35 detected]
[2] ESAM 3.0 SPE with SCTE 35 binary, zoneidentity, acquisitionPointIdentity
[3] ESAM 3.0 SPN either "noop" or "Content"
[4] Stay on stream or switch to alternate
[5] ESAM 3.0 SPN Response acknowledgement
[6] Appropriate linear stream
SAS
LRM DM (SDS)
vIRD

Figure 9 - Traditional ESAM Exchange

2.3.1. Message Transport

Messages are formatted and delivered in accordance with the CableLabs ESAM I03 specification. (CableLabs ESAM) The signal processing services are exposed as an HTTP RESTful endpoint. HTTP POST is the required request method and XML is the required message format, so the Content-Type header is set to application/xml. The SAS in this model is expected to return acknowledgements and processing status in the response.

2.3.2. Message Namespace

Table 2 shows the prefixes and namespaces that are used within the XML in ANSI/SCTE 35.

Table 2 - Example ANSI/SCTE 35 Namespaces

adi3	urn:cablelabs:md:xsd:core:3.0
common	urn:cablelabs:iptvservices:esam:xsd:common:1
content	urn:cablelabs:md:xsd:content:3.0
manifest	urn:cablelabs:iptvservices:esam:xsd:manifest:1
offer	urn:cablelabs:md:xsd:offer:3.0
po	urn:cablelabs:md:xsd:placementopportunity:3.0
signal	urn:cablelabs:iptvservices:esam:xsd:signal:1
signaling	urn:cablelabs:md:xsd:signaling:3.0
terms	urn:cablelabs:md:xsd:terms:3.0
title	urn:cablelabs:md:xsd:title:3.0
xsi	urn:cablelabs:iptvservices:esam:xsd:signal:1

2.3.3. Message Time

The CableLabs ESAM I03 specification requires that all object time values follow the ISO 8601 time format, hh:mm:ss, with an optional decimal fraction on the seconds component, e.g. 14:15:03.475. The specification also dictates that times are to be provided as zero UTC offset. All times in this document are zero UTC offset and may contain fractional seconds, as per the specification.

2.3.4. ESAM Interface

The ESAM interface supports signal confirmation. The ESAM interface carries ANSI/SCTE 35 signaling information from an SAS to the SDS for in-band processing. The originating system generates a processing signal event and submits it using the defined XML payload to the SDS. The event parameters provide as much information as possible about the signal point. Examples of in-band signal points are a splice out/exit point indicated by an ANSI/SCTE 35 splice_info_section()/splice_insert() command or an SCTE 35 segmentation_descriptor associated with a SCTE 35 time_signal() or splice_null() command. Using the provided information, the SDS derives information about the event start and end point(s) with the corrected timestamp and/or ANSI/SCTE 35 point data. Finally, the SDS generates a response notification detailing how the acquisition system might condition the video content and provides auxiliary data for downstream usage.

2.3.5. In-band ESAM Message Exchange

A request via the SAS utilizes the SignalProcessingEvent message as a wrapper to the AcquiredSignal element. The acquisitionPointIdentity attribute is used to correlate the stream to the proper source in the signal processor. The content of the AcquiredSignal element is a ANSI/SCTE 35 descriptor and is sent as a binary payload. The binary payload is Base64 encoded.

2.4. Use Cases

2.4.1. Decision Request

Upon detection of a ANSI/SCTE 35 marker, the SAS will assemble the ESAM Signal Processing Event (SPE) (Figure 10). For the SPE, the acquisitionPointIdentity must be set to the source of the video stream (i.e., KXYZ). The acquisitionSignalID is a uuid established by the SAS for correlating the SPE with other responses for tracking and logging. The acquisitionTime like acquisitionPointIdentity is required as well and represents the time the linear stream stitcher (LSS) created the SPE message. The last required element is zoneIdentity used to provide the name of the virtual integrated receiver decoder (vIRD) or station instance for matching against the ANSI/SCTE 224 audiences (i.e., WXYZ). Because ESAM does not provide a mechanism to differentiate between vIRDs and zipcodes, the SDS will use any five-digit numbers as a ZIP type audience. The UTCPoint represent the UTC of the signal in the video stream. Finally, the BinaryData element contains the binary data from the ANSI/SCTE 35 marker found in the video stream.

```
<esam:SignalProcessingEvent xmlns:esam="urn:cablelabs:iptvservices:esam:xsd:signal:1"
       xmlns:core="urn:cablelabs:md:xsd:core:3.0"
       xmlns:md="urn:cablelabs:md:xsd:signaling:3.0"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="urn:cablelabs:iptvservices:esam:xsd:signal:1 OC-SP-ESAM-API-I03-
Signal.xsd">
   <esam:AcquiredSignal
      acquisitionPointIdentity="CH1"
      acquisitionSignalID="3c1e5cbe-648a-4d69-b740-570a99cb66d7"
      acquisitionTime="2018-04-05T00:00:04Z"
      zoneIdentity="KXYZ">
      <md:UTCPoint utcPoint="2018-04-04T20:00:06.281Z"/>
      <md:BinaryData
signalType="SCTE35">/DBKAAAAAAAAAP/wBQb+u6m/DgA0AjJDVUVJAAAAAX//AARgYLMBHjAwMDM1TUEwMDAwMDAwMzE0MDdUMD
QwNTE4MDAzMBABAB4s/do=</md:BinaryData>
```

```
    </esam:AcquiredSignal>
</esam:SignalProcessingEvent>
```

Figure 10 - Example ESAM SignalProcessingEvent

2.4.2. *Decision Response*

Once the SDS receives the ESAM decision request, it will execute the SCTE 224 logic described above and return to the SAS the action required based on the ViewingPolicys. The response will contain the action of the source to switch to or “noop,” indicating the SAS does not need to switch from the source it is currently on.

Figure 11 shows the response the SAS will receive for no switch:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<signal:SignalProcessingNotification
    xmlns:adi3="urn:cablelabs:md:xsd:core:3.0"
    xmlns:signaling="urn:cablelabs:md:xsd:signaling:3.0"
    xmlns:signal="urn:cablelabs:iptvservices:esam:xsd:signal:1"
    xmlns:manifest="urn:cablelabs:iptvservices:esam:xsd:manifest:1"
    xmlns:ns5="http://www.cablelabs.com/namespaces/metadata/xsd/confirmation/2"
    xmlns:common="urn:cablelabs:iptvservices:esam:xsd:common:1"
    xmlns:content="urn:cablelabs:md:xsd:content:3.0"
    xmlns:offer="urn:cablelabs:md:xsd:offer:3.0"
    xmlns:po="urn:cablelabs:md:xsd:placementopportunity:3.0"
    xmlns:terms="urn:cablelabs:md:xsd:terms:3.0"
    xmlns:title="urn:cablelabs:md:xsd:title:3.0">
    <common:StatusCode classCode="0"/>
    <signal:ResponseSignal action="noop" acquisitionPointIdentity="CH1" acquisitionSignalID="27c3bf78-
3f65-4f03-ac44-4f61de204991" acquisitionTime="2018-05-18T15:53:03Z">
        <signaling:UTCPoint utcPoint="2018-05-18T15:53:07.184Z"/>
        <signaling:BinaryData
signalType="SCTE35">/DBKAAAAAAAAAP/wBQb+m3lcMAA0AjJDVUVJAAAAAX//AAG2TxcBHjAwMDM0TUEwMDAwMDAwMzI2ODJUMD
UxODE4MTUwMAEBADzcPCU=</signaling:BinaryData>
    </signal:ResponseSignal>
</signal:SignalProcessingNotification>
```

Figure 11 - Example ESAM SignalProcessingNotification Noop

Figure 12 shows the response the SAS will receive for a switch:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<signal:SignalProcessingNotification
    xmlns:adi3="urn:cablelabs:md:xsd:core:3.0"
    xmlns:signaling="urn:cablelabs:md:xsd:signaling:3.0"
    xmlns:signal="urn:cablelabs:iptvservices:esam:xsd:signal:1"
    xmlns:manifest="urn:cablelabs:iptvservices:esam:xsd:manifest:1"
    xmlns:ns5="http://www.cablelabs.com/namespaces/metadata/xsd/confirmation/2"
    xmlns:common="urn:cablelabs:iptvservices:esam:xsd:common:1"
    xmlns:content="urn:cablelabs:md:xsd:content:3.0"
    xmlns:offer="urn:cablelabs:md:xsd:offer:3.0"
    xmlns:po="urn:cablelabs:md:xsd:placementopportunity:3.0"
    xmlns:terms="urn:cablelabs:md:xsd:terms:3.0"
    xmlns:title="urn:cablelabs:md:xsd:title:3.0">
    <common:StatusCode classCode="0"/>
    <signal:ResponseSignal action="noop" acquisitionPointIdentity="CH1" acquisitionSignalID="27c3bf78-
3f65-4f03-ac44-4f61de204991" acquisitionTime="2018-05-18T15:53:03Z">
        <signaling:UTCPoint utcPoint="2018-05-18T15:53:07.184Z"/>
        <signaling:BinaryData
signalType="SCTE35">/DBKAAAAAAAAAP/wBQb+m3lcMAA0AjJDVUVJAAAAAX//AAG2TxcBHjAwMDM0TUEwMDAwMDAwMzI2ODJUMD
UxODE4MTUwMAEBADzcPCU=</signaling:BinaryData>
```

```
        <signal:AlternateContent altContent="true" altContentIdentity="CH5" zoneIdentity="KXYZ"/>
    </signal:ResponseSignal>
</signal:SignalProcessingNotification>
```

Figure 12 - Example ESAM SignalProcessingNotification for a Switch

2.4.3. SAS Response Acknowledgement

Upon receiving the SDS SPN response, the SAS will return an ESAM acknowledgement indicating the action taken by the SAS in the notes section. For the acknowledgement there are no specific requirements for the StatusCode::classCode for the ESAM-endpoint. The status code is only logged, but not used directly. For the Note element, there are also no specific requirements for its values. SDS logs this value, and currently SDS sees three common responses: “Successfully switched to CHXX,” “Cannot switch, because already scheduled to switch to that channel,” and “No switch, staying on CHXX” (Figure 13).

```
<?xml version="1.0" encoding="UTF-8"?>
<esam:ProcessStatusNotification acquisitionPointIdentity="KXYZ_PHILADELPHIA" acquisitionSignalID=
“3c1e5cbe-648a-4d69-b740-570a99cb66d7” xmlns:core="urn:cablelabs:md:xsd:core:3.0"
xmlns:esam="urn:cablelabs:iptvservices:esam:xsd:common:1" xmlns:md="urn:cablelabs:md:xsd:content:3.0"
xmlns:sig="urn:cablelabs:md:xsd:signaling:3.0" xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:cablelabs:iptvservices:esam:xsd:common:1 OC-SP-ESAM-API-I03-Common.xsd ">
        <esam:StatusCode classCode="0" detailCode="0">
             <core:Note>No switched needed source is already active : CH5 </core:Note>
        </esam:StatusCode>
</esam:ProcessStatusNotification>
```

Figure 13 - Example ESAM SignalProcessingNotification Acknowledgement

3. Distribution Workflow

With the ongoing consumer demands for higher quality video, as well as the new C-band rules from the FCC, content providers need to strongly consider a transition to non-satellite delivery. The workflow is similar whether there is a broadcaster distributing their 12-16 hours of daily content to an affiliate station, or if it is an affiliate station distributing their 24 hours of content to a distribution partner. In both cases there is content that needs to get to a distribution partner to be played out to a consumer. This disctribution workflow can be done with or without in-band signaling. This section will discuss the differences presented by the presence or absence of in-band signaling an dhow either situation can be handled.

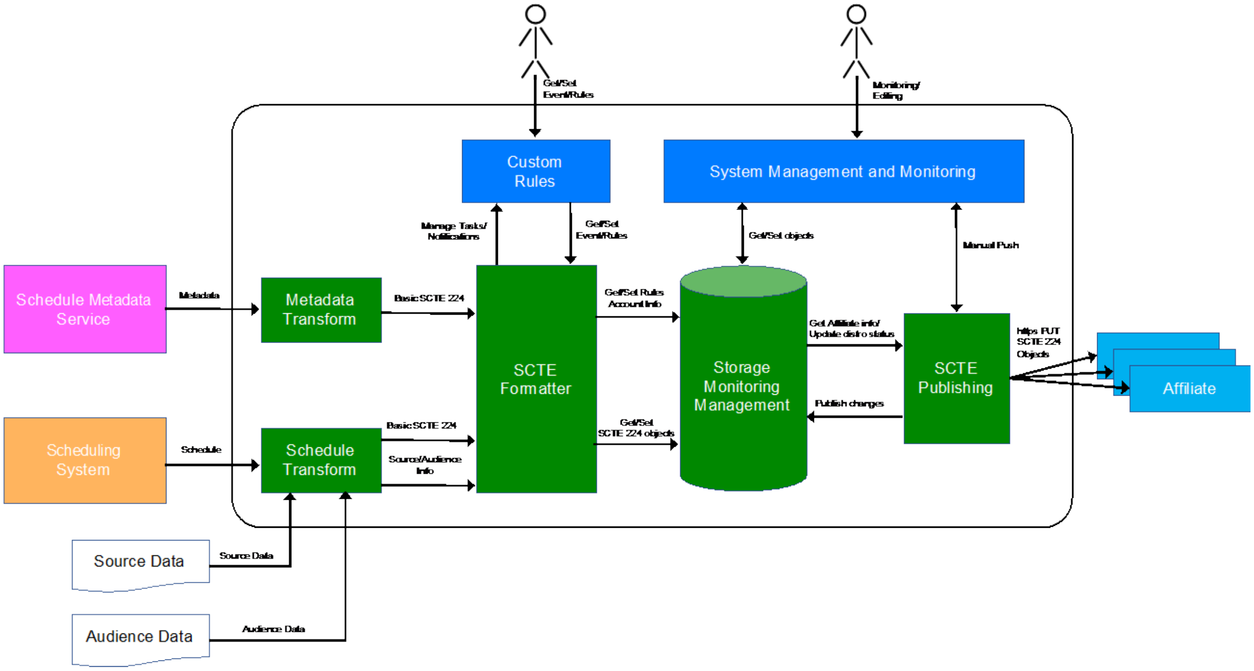


Figure 14 – Detailed Workflow

3.1. Transition with Automation

Many content providers use in-band signals even in the satellite feeds they are distributing today. This is most often the case for the full channel content being sent to multiprogram video program distributors (MVPD) or an OTT streaming service. This usually means that the content provider has an automation system capable of creating ANSI/SCTE 104 messages that are used to transfer signal insertion instructions from an automation system and convey those markings to the compression system. This ultimately results in creating the corresponding ANSI/SCTE 35 in-band markers for distribution in the video itself. If that is the case, and all program starts and ends are identified with ANSI/SCTE 35 binary signals, then in-band signals need not be injected into the video. Note that section 3.2 “Transition without Automation” describes the case where signals are missing and need to be injected.

3.1.1. Signaling

When in-band signals are sufficient to declare the content start and end, the burden on the content provider is to extract or correlate those ANSI/SCTE 35 UPIDs with the ANSI/SCTE 224 ESNI that is generated. These UPIDs must be extracted from the ANSI/SCTE 35 binary and placed in the “MatchSignal” element of the ANSI/SCTE 224 ESNI MediaPoint that goes along with that event. So, for example, if a system uses the show’s Tribune Media Service identifier (TMS ID) as the Upid to uniquely identify a program, then that TMS ID can be used by the ANSI/SCTE 224 generate platform to populate the XPath matching in the MediaPoint. Figure 15 shows an example of the MatchSignal portion of a ANSI/SCTE 224 MediaPoint.

```
<MatchSignal xmlns="http://www.scte.org/schemas/224" match="ANY">
    <Assert>
/SpliceInfoSection/SegmentationDescriptor[@segmentationTypeId=16]/SegmentationUpid[@segmentationUpidTy
pe=1 and contains(text(),'EP001786121816')]
    </Assert>
```

```
</MatchSignal>
```

Figure 15 - Example MediaPoint MatchSignal

Although automation inserts ANSI/SCTE 35 signals in the video stream, downstream automation needs to get Upid information (such as “EP001786121816” from the example above) to the ANSI/SCTE 224 ESNI creation system.

3.1.2. *Additional Instruction*

At this point the in-band signaling information is in a MediaPoint’s MatchSignal element. On playout, if the SAS received the in-band signal, the SDS would match on a MediaPoint. The SDS replies with the “Action” to be taken by the acquisition system, based on the audience of interest. This is done in ANSI/SCTE 224 by using “Apply” policy to activate a policy. The Policy then points to the ViewingPolicy, which has the real functionality of bringing an audience together with an action.

For a broadcast solution, where a content provider is selecting a feed for a particular show, the provider can identify the desired audience, for example by ZIP code or audience type. If the SAS identity matches the audience then the action applies. The action could indicate the desired content, such as from an alternate channel source, or a URL from which the content could be retrieved. With this model, a content provider could tell one partner to play content from one channel and another partner to play content from another.

For example, a content provider of a broadcast network can tell its Eastern time zone partners to pull content from one feed, while telling its Central time zone partners to pull from a different feed. Alternatively, for a live event, the content provider could tell all partners to source from the same live feed.

Likewise, a content provider distributing a channel could tell one partner to black out or display a slate during a particular program, while having another region or provider play the content on the current feed. This situation is useful for sports blackouts or when content rights vary between regions.

3.2. Transition without Automation

In many cases, content providers do not have sufficient in-band signaling in their video feeds, possibly because satellite feeds rarely needed detailed signaling, or because most or all of the controls are done by integrated receiver/decoders (IRDs). Another situation is that OTT feeds, or higher quality feeds that are sourced terrestrially, are not subject to the same workflow. Additionally, the automation system driving the satellite delivery might lack signaling features whatsoever. SCTE 224 ESNI accommodates for these cases and still allows signals to be inserted without the corresponding automation.

3.2.1. *Signal Insertion*

The ANSI/SCTE 224 ESNI standard provides actions in the ViewingPolicy to enable signal insertion. In this sense, the content provider delivers schedule information to the generator; the generator instructs the SAS to inject signals when needed. In this case, generator creates two sets of instructions in the system. The first set of Media contains MediaPoints for each show start and stop, but the action is a “SignalPointInsertion” action. The second set of Media are the same MediaPoints for the shows, but this Media contains the MatchSignal logic to connect to the signals that are being inserted. This second set of Media is for distribution to execute specific actions.

Figure 16 depicts the first Media, marking the distributed video for edge execution, and the second Media being used at the edge to execute the content provider's rights for the content.

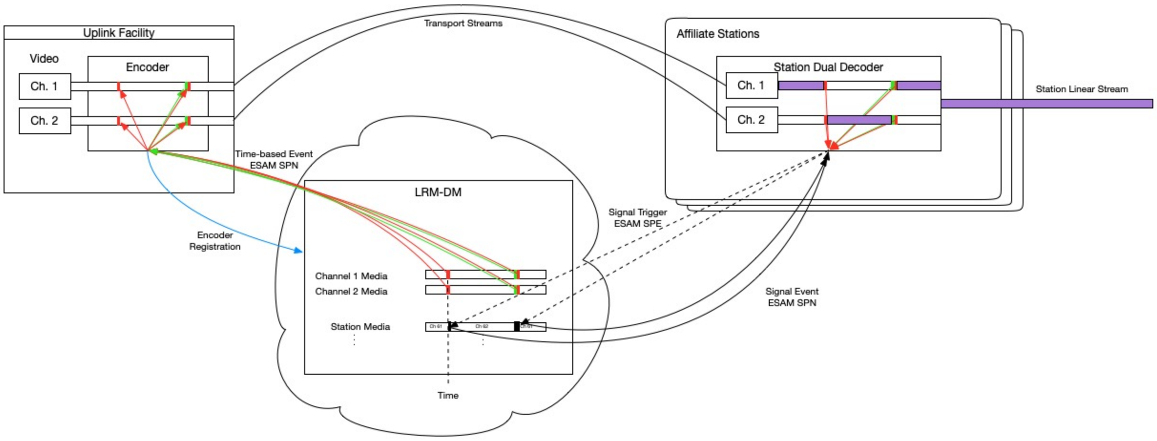


Figure 16 - Workflow for Signal Insertion

When the scheduling system sends the ANSI/SCTE 224 ESNI generator the schedule information, the generator has to know to generate two different Media elements. The start and end times are the same for both Media elements. The difference is the Policy that the MediaPoints point to, and that the MatchSignal element is only in one of the Media elements.

3.2.1.1. Signal Insertion Media

The signal insertion Media contains the audience of the channel into which the signals will be inserted. If the event will not be triggered by in-band signals, the insertion will be initiated by the SDS itself. These SDS-initiated ESAM communications are referred to as "unsolicited messages." In these cases, the MediaPoints in the containing Media element will have a matchtime attribute that represents the precise time that the event will occur, and the signal should be inserted.

In many SDS systems, a time-based, unsolicited message is coupled with an ANSI/SCTE 250 ESAM registration by the SAS. This registration tells the SDS which endpoint to send the unsolicited messages. It also usually contains lead-time information, about when to send the instructions, and, finally, the desired audience. For example, if a show starts on feed X at 2:00 PM, and an SAS registration is interested in receiving the signaling for feed X with a one minute lead time, then the SDS sends an unsolicited ESAM SignalProcessingNotification message at 1:59 PM to the SDS endpoint. This gives the SAS the time necessary to insert the signal at the proper time in the video.

Once this in-band signal has been inserted, the SAS will match it on the execution side using the other Execution Media.

3.2.1.2. Distribution Media

The second Media generated from the show's schedule has the same start and ends times, but in this Media the ESNI generator creates the MatchSignal elements and inserts the signal information used above. It also can populate the show's asset distribution interface (ADI) information if the content provider so desires, since this Media is used on the edge by the distribution partner.

3.2.1. *Additional Instruction*

This Distribution Media then follows the same execution path that the "Automation" section described, where the SAS detects the in-band signal and sends it to the SDS for a decision about what to do at the point in the video feed. The SDS then responds with the appropriate action for the audience represented by the SAS.

4. Solution

At Comcast Technology Solutions, we created a solution that bridges the creation, management and distribution process of ANSI/SCTE 224 events for both broadcast and OTT environments. The service automates the manual processes and takes full advantage of the ANSI/SCTE 224 specification – not just by standardizing the formatting of metadata, but also by providing a software-based service that gives both programmers and operators unprecedented control of – and visibility into – their respective workflows. As more variables are introduced into the content delivery mix, whether they be new content types or new viewing policy actions, we will always need a flexible strategy that can deliver extraordinary experiences across every screen. Providers can simplify processes and utilize better content management tools, and then apply them to what matters most: creating an experience that turns viewers into ardent and vocal fans.

5. Additional thoughts

5.1. OTT Complexity

Complex viewing rights have become an increasing challenge for content delivery. No matter what the specific rights are for a program -- whether they pertain to regional networks, league- or match-specific arrangements, geography or user permissions -- it has become crucial that providers are precisely conveying the policies necessary to allow for authorized playback. Digital distribution partners require these rights in order to distribute the client's content outside the home viewing territory. ANSI/SCTE 224 stands out as the best way to deliver this information. The new metadata management approach is now an integral part of operational workflows and is managed by an internal operations team.

The solution supports both MatchTime-based event rules and MatchSignal-based event rules. This means a full functional adaptation of ANSI/SCTE 35 segmentation UPID, or signal ID's, at the ProgramStart and ProgramEnd boundaries of a MediaPoint, and is able to map the ANSI/SCTE 35 ID to the MatchSignal ID in the ANSI/SCTE 224 event. A clean transition is ensured when alternate content rules are applied to the linear feed. When an event window runs beyond the scheduled airing time, the alternate content execution is based on the segmentation UPID identified in the video feed, matching the UPID that is in the ANSI/SCTE 224 event. This allows for precision control from the content provider, while maintaining a positive user experience for the viewer.

Linear rights data can now be distributed as many as 14 days ahead of the airing of the event, and real-time distribution solves for event changes, such as a baseball game that goes into extra innings. As a broadcaster that distributes hundreds of regional sports networks, managed at a national level while being distributed by digital distribution partners or OTT services, solving for unscheduled event changes is important. Using a push model, linear programing events are distributed to the metadata management tool. Operations teams now have visibility into the program schedule for all channels, because that data is normalized into ANSI/SCTE 224 and then distributed to all digital distribution partners. Through a desktop console, operations teams can:

- Make real-time changes to linear events
- View programming schedules
- Edit and create audience definitions
- Perform policy updates
- Validate decision logic and audit activity

Following the ANSI/SCTE 224 standard, all events are distributed using XML over HTTP to designated URL endpoints. Messages have an option to be signed using an assigned key value, and have a configurable frequency as well as a retry rate. The messages can also be individually targeted to each distribution partner and filtered, so each partner gets only what they need.

As an organization among the early adopters of ANSI/SCTE 224, it's worth noting that several of the largest broadcasters, at the forefront of the industry, are using these extant standards and resultant tools to take the lead on how linear rights metadata can and should be delivered.

5.2. EPG Usage

An additional benefit of transitioning to ANSI/SCTE 224 ESNI is ease in sharing data used to populate EPGs. Once broadcasters have an OTT schedule completely decorated in ESNI, then that schedule can be shared with an OTT partner, who can process the information in a separate workflow for the EPG. For example, if the ESNI schedule is populated for seven days in the future, then an OTT distribution partner can also populate its EPG into the future. Unlike the generic program guide information, an ESNI schedule contains alternate content information. Thus, the partner can present the subscriber with a highly accurate, tailored EPG. They may then look ahead in the schedule and see the "real" guide for the specific experience.

Likewise, if the content provider is making changes to the playout schedule, the corresponding ANSI/SCTE 224 ESNI schedule changes immediately. It flows to the distribution partner instantaneously, and faster than third-party services that manage program guide data.

6. Conclusion

As broadcasters adapt their access to and use of the C-band spectrum for traditional video distribution, concurrent with the FCC's reallocation of the lower 280 MHz of that band for 5G wireless service, many are opting to shift to terrestrial, fiber-based delivery. It's a compelling shift, with benefits that include lower costs, lower latencies, fewer bandwidth constraints, to name but a few.

Any transition to terrestrial video delivery over fiber must occur with minimal impact to current customer workflows. The risks of transition certainly include disruption of primary service, but also of signaling for advertising, local contributions, and alternate programming, including blackouts. The benefits include additional OTT distribution. Certain hybrid use cases increase disruption risk. For example, national feeds arriving by satellite can signal a local contribution arriving over fiber, or vice versa.

The SCTE DDVS maintains a family of standards that, when implemented together, help support a seamless transition from satellite to fiber. These standards can be applied, in real-world situations, to enable a transition to terrestrial, fiber-based delivery with minimal impacts to workflows.

This paper detailed an architecture and specifications for a linear content signaling and decision system that is useful for implementing the intricacies of rights management, affiliate programming, and satellite delivery replacement. It detailed the use of ANSI/SCTE-35, ANSI/SCTE-224 (ESNI) and ANSI/SCTE-250 (ESAM) into an architecture useable by on-premise and cloud-based systems.

Comcast Technology Solutions has developed and deployed this system for hundreds of channels and thousands of decisions per week. Implementing this linear system enables the seamless transition of fully featured linear channels, from satellite to the Internet.

Abbreviations

ANSI	American National Standards Institute
API	application programmer interface
ADI	asset distriution interface
dMVPD	digital multichannel video programming distributors
DVS	[SCTE] Digital Video Subcommittee
EPG	electronic program guide
ESAM	Event Signaling and Management
ESNI	Event Scheduling and Notification Interface
FCC	Federal Communications Commission
HTTP	Hypertext Transfer Protocol
IRD	integrated receiver decoder
ISBE	International Society of Broadband Experts
ISO	International Standards Organization
LSS	linear stream stitcher
MPEG	Moving Pictures Experts Group
MVPD	multichannel video programming distributors
OTT	over-the-top
REST	Representational State Transfer
SAS	signal acquisition service
SCTE	Society of Cable Telecommunications Engineers
SDS	signal decisioning service
SPE	signal processing event
SPN	signal processing notification
TMS	Tribune Media Service
UPID	unique program identifiers
URL	Uniform Resource Locator
UTC	Universal Time Coordinated
vIRD	virtual integrated receiver decoder
XML	Extensible Markup Language

Bibliography & References

CableLabs ESAM, Real-time Event Signaling and Management API, 2013. https://scte-cms-resource-storage.s3.amazonaws.com/SCTE 35 OC-SP-ESAM-API-I03-131025.pdf.

FCC Report and Order on C-band. Expanding Flexible Use of the 3.7 to 4.2 GHz Band https://docs.fcc.gov/public/attachments/FCC-20-22A1.pdf

Reardon, M. February 6, 2020. FCC reaches deal with satellite industry to free up more 5G spectrum, CNet, retrieved from: https://www.cnet.com/news/fcc-reaches-deal-with-satellite-industry-to-free-up-more-5g-spectrum/

SCTE 224 ESNI, Event Scheduling and Notification Interface 2018r1. https://scte-cms-resource-storage.s3.amazonaws.com/Standards/ANSI_SCTE%20224%202018r1.pdf

SCTE 35, Digital Program Insertion Curing Message for Cable, 2019. https://scte-cms-resource-storage.s3.amazonaws.com/ANSI_SCTE-35-2019a-1582645390859.pdf

Appendix A

Example ANSI/SCTE 224 Media Object

```
<Media xmlns="http://www.scte.org/schemas/224"
 id="xyz/media/WXYZ"
 description="WXYZ"
 lastUpdated="2020-03-24T17:49:45.000Z">
  <MediaPoint xmlns="http://www.scte.org/schemas/224"
   id="xyz/media/WXYZ/resident"
   description="WXYZ"
   lastUpdated="2020-03-24T17:49:45.000Z"
   effective="2020-03-23T18:00:00.000Z"
   expires="2020-04-01T08:30:00.000Z">
    <Apply xmlns="http://www.scte.org/schemas/224">
      <Policy xmlns="http://www.scte.org/schemas/224"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xlink:href="xyz/policy/any/slate"></Policy>
    </Apply>
  </MediaPoint>
  <MediaPoint xmlns="http://www.scte.org/schemas/224"
   id="xyz/media/WXYZ/program/3E264A41-8BE7-41D9-8FAF-2072EFA0A866/start"
   description="Show 1"
   lastUpdated="2020-03-24T17:49:45.000Z"
   effective="2020-03-23T17:30:00.000Z"
   expires="2020-03-23T19:30:00.000Z"
   source="3230">
    <AltID xmlns="http://www.scte.org/schemas/224">EP002830833983</AltID>
    <Metadata xmlns="http://www.scte.org/schemas/224">
      <MetadataDetail xmlns="http://ctsrmm.com/ctsesni"
        name="ScheduledAiringId"
        type="string"
        provider="XYZ">3E264A41-8BE7-41D9-8FAF-2072EFA0A866</MetadataDetail>
      <MetadataDetail xmlns="http://ctsrmm.com/ctsesni"
        name="ScheduledStart"
        type="string"
        provider="XYZ">2020-03-23T18:00:00.000Z</MetadataDetail>
      <MetadataDetail xmlns="http://ctsrmm.com/ctsesni"
        name="ScheduledEnd"
        type="string"
        provider="XYZ">2020-03-23T19:00:00.000Z</MetadataDetail>
      <MetadataDetail xmlns="http://ctsrmm.com/ctsesni"
        name="StartOver" type="string"
        provider="XYZ">false</MetadataDetail>
      <MetadataDetail xmlns="http://ctsrmm.com/ctsesni"
        name="LookBack"
        type="string"
        provider="XYZ">false</MetadataDetail>
      </MetadataDetail>
      <ADI3 xmlns="http://www.scte.org/schemas/236/2017/core">
        <Asset xmlns:XMLSchema-instance="http://www.w3.org/2001/XMLSchema-instance"
      XMLSchema-instance:type="title:TitleType"
           uriId="xyz.com/Title/3E264A41-8BE7-41D9-8FAF-2072EFA0A866"
           startDateTime="2020-03-27T14:14:40Z"
           endDateTime="2020-03-27T14:30:16Z"
           lastModifiedDateTime="2020-03-27T15:04:46.744Z">
          <Provider>XYZ</Provider>
          <Ext>
            <App_Data Name="Season_Number" Value="1"></App_Data>
            <App_Data Name="Episode_Number" Value="S1:E14"></App_Data>
            <App_Data Name="Series_Name" Value="Show 1"></App_Data>
            <App_Data Name="Episode_Title" Value="Show 1"></App_Data>
            <App_Data Name="Season_Title" Value="Show 1"></App_Data>
            <App_Data Name="Category" Value="Drama"></App_Data>
            <App_Data Name="Is_Live" Value="N"></App_Data>
            <App_Data Name="DVS" Value="N"></App_Data>
          </Ext>
          <LocalizableTitle xmlns="http://www.scte.org/schemas/236/2017/title">
            <TitleBrief xmlns="http://www.scte.org/schemas/236/2017/title">Maury</TitleBrief>
            <TitleMedium xmlns="http://www.scte.org/schemas/236/2017/title">Maury</TitleMedium>
            <TitleLong xmlns="http://www.scte.org/schemas/236/2017/title">Maury</TitleLong>
            <SummaryShort xmlns="http://www.scte.org/schemas/236/2017/title">Funny
Story</SummaryShort>
          </LocalizableTitle>
          <Rating xmlns="http://www.scte.org/schemas/236/2017/title"
             ratingSystem="TV">TV-Y7-FV</Rating>
          <IsClosedCaptioning xmlns="http://www.scte.org/schemas/236/2017/title">
             true</IsClosedCaptioning>
          <DisplayRunTime xmlns="http://www.scte.org/schemas/236/2017/title">
             00:15</DisplayRunTime>
          <Year xmlns="http://www.scte.org/schemas/236/2017/title">2019</Year>
          <ShowType xmlns="http://www.scte.org/schemas/236/2017/title">Series</ShowType>
        </Asset>
        <Asset xmlns:XMLSchema-instance="http://www.w3.org/2001/XMLSchema-instance"
          XMLSchema-instance:type="content:MovieType"
          uriId="xyz.com/Asset/3E264A41-8BE7-41D9-8FAF-2072EFA0A866"
          providerVersionNum="10"
          internalVersionNum="0"
          creationDateTime="2020-03-26T22:20:19.762Z"
          startDateTime="2020-03-27T14:14:40Z"
          endDateTime="2020-03-27T14:30:16Z"
          lastModifiedDateTime="2020-03-27T15:04:46.744Z">
         <Provider>XYZ</Provider>
         <AudioType xmlns="http://www.scte.org/schemas/236/2017/content">Dolby 5.1</AudioType>
         <Language xmlns="http://www.scte.org/schemas/236/2017/content"
           bitStreamMode="0">en</Language>
        </Asset>
      </ADI3>
    </Metadata>
    <Apply xmlns="http://www.scte.org/schemas/224">
      <Policy xmlns="http://www.scte.org/schemas/224"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xlink:href="xyz/policy/WXYZ.all/3230">
      </Policy>
    </Apply>
    <MatchSignal xmlns="http://www.scte.org/schemas/224" match="ANY">
      <Assert xmlns="http://www.scte.org/schemas/224">
        /SpliceInfoSection/SegmentationDescriptor[@segmentationTypeId=16]/
        SegmentationUpid[@segmentationUpidType=1 and contains(text(),’EP002830833983’)]</Assert>
      <Assert xmlns="http://www.scte.org/schemas/224">
        /SpliceInfoSection/SegmentationDescriptor[@segmentationTypeId=1]/
        SegmentationUpid[@segmentationUpidType=1 and contains(text(),’EP002830833983’)]</Assert>
    </MatchSignal>
  </MediaPoint>
  <MediaPoint xmlns="http://www.scte.org/schemas/224"
     id="xyz/media/WXYZ/program/3E264A41-8BE7-41D9-8FAF-2072EFA0A866/end"
     description="Show 1"
     lastUpdated="2020-03-24T17:49:45.000Z"
     effective="2020-03-23T18:30:00.000Z"
     expires="2020-03-23T19:30:00.000Z"
     source="3230">
    <Remove xmlns="http://www.scte.org/schemas/224">
      <Policy xmlns="http://www.scte.org/schemas/224"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xlink:href="xyz/policy/WXYZ.all/3230">
      </Policy>
    </Remove>
    <MatchSignal xmlns="http://www.scte.org/schemas/224" match="ANY">
      <Assert xmlns="http://www.scte.org/schemas/224">
        /SpliceInfoSection/SegmentationDescriptor[@segmentationTypeId=17]/
        SegmentationUpid[@segmentationUpidType=1 and contains(text(),’EP002830833983’)]</Assert>
      <Assert xmlns="http://www.scte.org/schemas/224">
        /SpliceInfoSection/SegmentationDescriptor[@segmentationTypeId=1]/
        SegmentationUpid[@segmentationUpidType=1 and
        not(contains(text(),’EP002830833983’))]</Assert>
    </MatchSignal>
  </MediaPoint>
```

```
</Media>
```