WHITE PAPER

A Simple Description of SDP in SMPTE ST2110

Authors :

Renaud Lavoie
Sithideth Viengkhou
Sébastien Berthiaume

INTRODUCTION

During our discussions with customers about SMPTE ST2110 and media-over-IP, we are often asked questions about receivers and in particular, how they understand the nature of the IP traffic they receive in SMPTE ST2110. In this whitepaper, we will set out to answer that question by presenting a high-level overview of SDP (Session Description Protocol), which ensures that receivers know the nature of all flows arriving through the IP port.

As a starting point, it is important to understand where the SDP protocol is located within the OSI (Open Systems Interconnection Model est.1983 by the International Organization for Standardization)[1]. See figure 1.

As figure 1 illustrates, we can see that the SDP protocol is found on the Application Layer, along with the SIP (Session Initiation Protocol) and SAP (Session Announcement Protocol). Video, audio and metadata flows are located on the Transport Layer. SDP is communicated within the SIP framework and, as an application, configures flows on the transport layer.

THIS PAPER EXPLORES:

- The importance of SDP
- The SDP Exchange
- Possible Improvements
- Examples of SDP

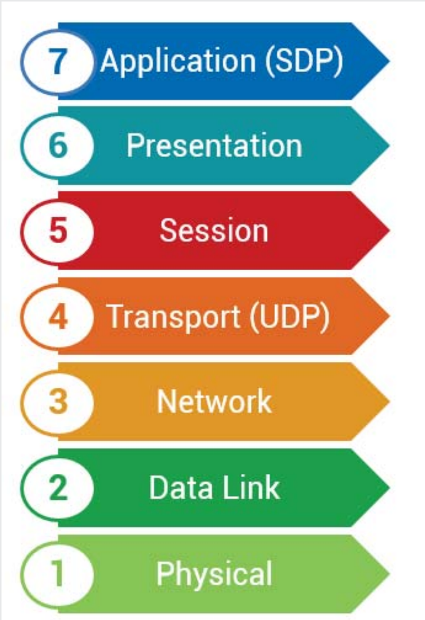

Figure 1. Open Systems Interconnection Model Illustrating SDP

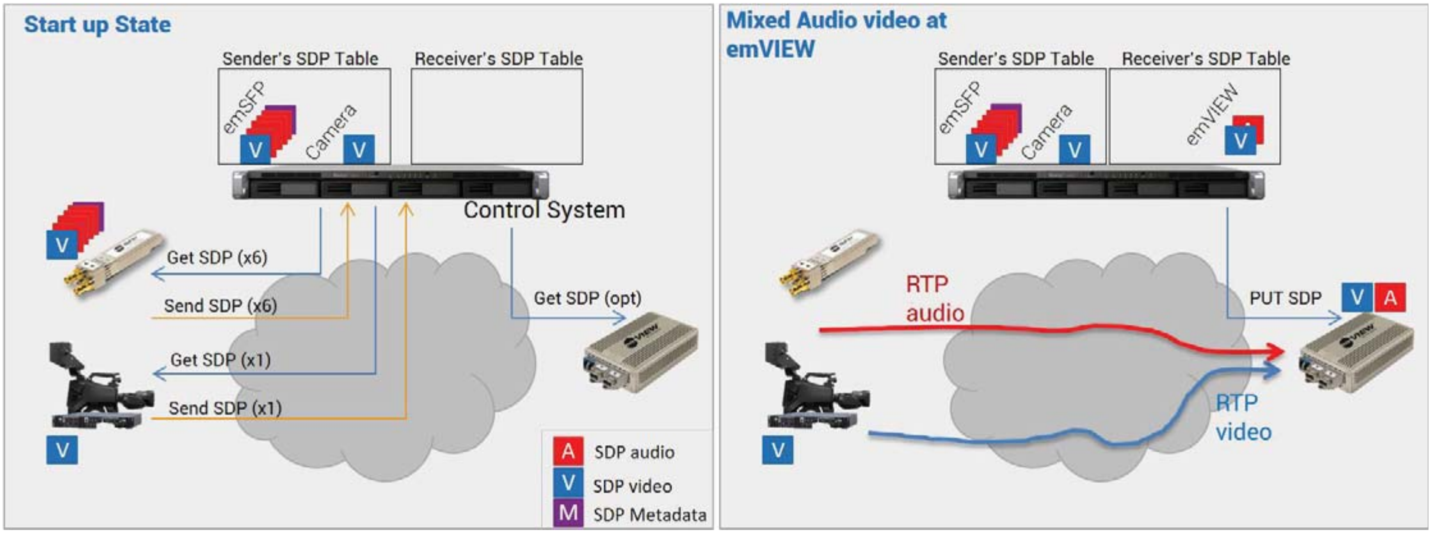

Figure 2. Example of (Control System) Acquiring SDP from Sender and Assigning SDP to Receiver

ABOUT SDP

As mentioned, SDP is a protocol that describes communications parameters for streaming media for the purpose of session announcement, session invitation, and parameter negotiation[2].

SDP doesn't deliver media per se, but is rather "used between end points for negotiation of media type, format, and all associated properties, called session profile."[3] In other words, SDP provides the information between senders and receivers in multiple applications, and primarily for media applications, including mobile networks.

In 1998, the SDP specification was first published by the IETF (Internet Engineering Task Force). A revised version was then available in July, 2006 as a proposed standard called RFC 4566[4]. This revised standard was selected by the SMPTE Working Group[5] for the broadcast market and evolved with SAP and SIP in the early 2000s. The following sections explain the functionality of the SDP protocol in broadcast.

THE IMPORTANCE OF SDP IN SMPTE ST2110

In SMPTE ST2110, essences are separated in the sender and sometimes flows come from different senders. It is also important to remember that flows contain intrinsic information such as video format, frame rate, colorimetry, etc.

Within a SMPTE ST2110 installation, all stream description is made available via the control system (and in the case of Embrionix products, is available through the Restful API). The control application acquires the desired stream's SDP file in order to read the provided information description (see figure 2 above).

- In SMPTE ST2110, the control system acquires the SDP file from the sender and sends it to the receiver. One SDP file is present for each flow. If you are using Embrionix products in non-hitless mode, the sender will have six (6) SDP files: one (1) for the video flow, four (4) for the audio flows and one (1) for the metadata flow.
- The SDP file contains both Amber and Blue network APIs when used in a redundant system using 2022-7 implementation. The SDP file will include details for streams sent to NET 1 (Amber) and NET 2 (Blue).

In the start up state, (the initial moment where the equipment is started after the installation of the system), the control system will discover the devices and fetch the SDP files from senders (receivers are optional).

To acquire the SDP file, the control system can use the UID to obtain the associated SDP files from the flows. The control system will then build the SDP table (which will include the SDP file information for each device) for each sender. Again, receivers are optional.

SDP allows a constant update to occur between the sender and the receiver

The second image in Figure 2, shows the control system using a PUT to the Embrionix emVIEW gateway (receiver) to assign the two flows – video arriving from the camera and audio arriving from the Embrionix emSFP encapsulator.

- Note that video/audio/medatada are typically transported on the same physical network called the media network. Control can also be transported inside the media network -as in-band network - or it can be sent over a different network typically called control network
- Note that it is the control system's responsibility to maintain an up-to-date SDP. For example, if the format of the source of the encapsulator changes, the control system should implement a mechanism to obtain the new SDP from the encapsulator

POSSIBLE IMPROVEMENT TO THE SDP CONNECTION METHOD

One drawback related to the SDP connection method concerns the configuration of the sender and tha fact that it can change without notice. This makes it impossible for the receiver to understand when something has changed on the sender in order to re-adapt in consequence. However, SAP allows a constant update process between the two.

As mentioned, SDP evolved with SAP and under SAP, senders periodically broadcast SDP description to a well-known multicast address and port. Either a SAP listening application or receivers can acquire SAP multicasts and construct SDP tables (for senders) and this ensures that SDP files are always up to date.

This being said, a constant flow of updates on the network can cause an overflow of traffic. The interval (time by which the file is updated to the receiver) is defined to a maximum of 300 seconds (five minutes). By default, SAP consumes a maximum of 4000 bits per second, which is negligible compared to the bandwidth of video signals in the network.

- Per figure 3, below, when allowing for SAP-enabled devices, any endpoints (receiver in this case) can listen to a multicast SDP and adapt behavior if the sender's SDP information changes

In figure 3, the Embrionix emSFP signal format changes from 1080i to 4K, and then the Embrionix emVIEW gateway changes the format received and the HDMI output to 4K with no need for intervention by the control system.

There is room for additional improvement (on SAP-enabled sender devices) when sending the SDP multicast asynchronously in times where there is a change of input format. Currently the control system monitors the sender signal format and sends the SDP file to the receiver if a change occurs. In addition, the specification does not include a time to switch information. This could be used by a receiver to accurately switch at a specific time between two senders in a synchronous manner

Embrionix products have been tested with SAP activated and no degradation or jitter has been observed on the video/audio or metadata payloads.

SDP EXCHANGE IN SMPTE ST2110

A general SDP exchange involves: the sender flow being read via a "GET" in order to read the SDP file. A simple "PUT" to the receiver with the sender's SDP as the message body, will then connect the sender's flow to the receiver. The receiver oversees handling the validation of the sender's SDP information; including whether the format from the sender is compatible.

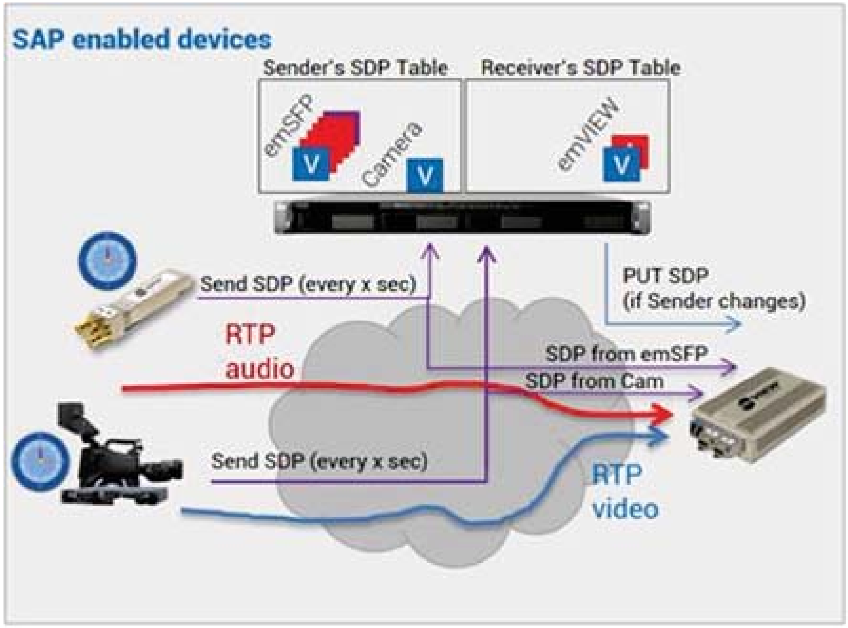

Figure 3. SAP Functionality

SDP - EXAMPLE

The goal of the SDP file is to allow a sender device to describe, with sufficient information, the content or format of the sent data in such a way as to allow the receiving device to properly adapt to the data. Below are tables that can help us to better understand SDP file fields.

Table 1. General SDP File Fields

Field	Description
Version	V=0
Origin	O= <username> <session id> <version> <network type> <address type> <address>
Session Name	S= <session name>
Times	T=<start time> <stop time>
Connection Data	C=<network type> <address type> <connection address>
Media	M=<media> <port> <transport> <media format list>

Table 2. Single Flow SDP Example

Version	V=0
Origin	O=- 1443716955 1443716955 IN IP4 10.194.205.201
Session Name	S=st2110 stream
Times	T=0 0
Connection Data	C=IN IP4 225.17.2.2/64
Media	M=video 20000 RTP/AVP 0
Attribute source filter	a=source-filter:incl IN IP4 225.17.2.2 10.194.205.102
Attribute dynamic payload	a=rtpmap:0 smpte291/90000
Attribute media format	a=fmtp:0 VPID_Code=133
Attribute media clock	a=mediaclk:direct=0 rate=90000

Table 3. Dash-7 flow SDP example

Version	V=0
Origin	O=- 1443716955 1443716955 IN IP4 192.168.39.15
Session Name	S=st2110 stream
Times	T=0 0
Atribute group identification	a=group:DUP primary secondary
Media	M=video 20000 RTP/AVP 96
Connection Data	C=IN IP4 239.0.1.2/64
Attribute source filter	a=source-filter:incl IN IP4 239.0.1.2 192.168.0.1
Attribute dynamic payload	a=rtpmap:96 raw/90000
Attribute media format	a=fmtp:96 sampling=YCbCr-4:2:2; width=1920; height=1080; exactframerate=30000/1001; depth=10; TCS=SDR; colorimetry=BT709; PM=2110GPM; SSN="ST2110-20:2017"; TP=2110TPN; interlace=1
Attribute media clock	a=mediaclk:direct=0
Attribute PTP reference clock	a=ts-refclk:ptp=IEEE1588-2008:08-00-11-ff-fe-22-91-bb:0
Attribute group section	a=mid:primary
Media	m=video 20000 RTP/AVP 96
Connection data	c=IN IP4 239.0.1.3/64
Attribute source filter	a=source-filter:incl IN IP4 239.0.1.3 192.168.0.1
Attribute dynamic pyload	a=rtpmap:96 raw/90000
Attribute media format	a=fmtp:96 sampling=YCbCr-4:2:2; width=1920; height=1080; exactframerate=30000/1001; depth=10; TCS=SDR; colorimetry=BT709; PM=2110GPM; SSN="ST2110-20:2017"; TP=2110TPN; interlace a=mediaclk:direct=0
Attrbiute PTP reference clock	a=ts-refclk:ptp=IEEE1588-2008:08-00-11-ff-fe-22-91-bb:0
Attribute group section	a=mid:secondary

CONCLUSION

This paper endeavored to offer a a high-level overview of SDP (Session Description Protocol), which is used to ensure that receivers know the nature of all flows arriving through the IP port and the importance of SDP in SMPTE ST2110. Through exploring SDP exchanges in SMPTE ST2110 and examples of SDP, we discover its utility as well as possible improvements to the SDP connection method.

For more information please contact: info@embrionix.com/sales@embrionix.com

-Source 1: Encyclopedia Britannica (https://www.britannica.com/technology/open-systems-interconnection)

-Source 2, 3, 4 & 5: Wikipedia (https://en.wikipedia.org/wiki/Session_Description_Protocol)

embrionix

2120, rue Monterey, suite 200, Laval (Quebec) H7L 3S3 Canada

450-688-8171