

Tutorial: How to Send and Receive SRT Streams With Captions Using iCap Alta

Video delivered over the public internet took a big step forward when SRT (Secure Reliable Transport) arrived. An open-source video transport and technology stack, SRT has proven key to carrying secure, high-quality video to devices for several years.

The newly released SRT 1.5, [announced at IBC 2022](#), further improved the protocol's decentralized capabilities, stability and reliability for broadcast production, remote contribution, and content distribution. While SRT's ability to optimize streaming performance across unpredictable networks is well known, its captioning workflows don't always receive as much attention.

To make things easier for SRT users, Ai-Media and EEG present a trio of tutorials for sending and receiving SRT streams with captions added with the [iCap Alta Software Caption Encoder for Live IP Video](#). iCap Alta is ideal for content creators adding closed captioning/subtitling to IP video production environments. We start with the OBS (Open Broadcaster Software) platform, and then cover additional variations.

Read on and learn how to caption

- Using OBS as a streaming source
- Using another streaming source other than OBS
- An SRT source stream that can go directly into Alta

FLOW 1: Using OBS as Streaming Source

This flow shows how SRT streams can be created, encoded, and viewed in real time when using OBS as your streaming tool. In this case, the stream will have the following path:

OBS > MediaLive > MediaConnect > Alta > MediaConnect > VLC

When using OBS as the streaming tool for creating a source stream (it creates RTMP input streams), you would:

1. Send the stream from OBS to MediaLive. This is done to convert it from RTMP to an input type supported by the [AWS Elemental MediaConnect](#) high quality live video transport service (such as UDP, RTP, or http), as MediaConnect does not support RTMP.
2. From MediaLive, the converted stream gets sent to MediaConnect, which will then convert the stream again to SRT after selecting SRT as the output protocol (MediaLive is unable to convert to SRT, which is why this is necessary).
3. Then, the SRT stream can be sent directly into Alta, where the stream will get encoded with captions.

4. From Alta, you can then send the output SRT stream back to MediaConnect in order to be able to send it from there to VLC media player, where the created SRT stream can be viewed live in real-time.

Here's how to accomplish the above steps, in greater detail

1. **Set up an SRT stream** that is sourced from OBS is to create a channel in [AWS Elemental MediaLive](#) (a broadcast-grade live video processing service that creates high-quality streams). There are several steps involved in creating a MediaLive channel, most of which are beyond the scope of this article, but there are several things important to our workflow:

Since the input stream is being sourced from OBS, RTMP will be the input type, so the MediaLive Channel should have an Input Attachment of type **RTMP_PUSH**

AWS Elemental MediaLive > Inputs > Create input

Create input

Input details

Input name – *required*

Input type – *required*

- RTP
Push your source to fixed endpoints with the real-time transport protocol.
- RTMP (push)
Push your source to fixed endpoints with the real-time messaging protocol.
- RTMP (pull)
Pull your source from external endpoints with the real-time messaging protocol.
- HLS
Pull your source from external endpoints with the HTTP protocol.
- MP4
Ingest file content from an MP4 file that is on the public internet
- TS
Ingest transport stream content from a TS file from external endpoints specified in the url.
- MediaConnect

Output Delivery from MediaLive should be VPC (Virtual Private Cloud) since it's being sent out to another AWS service (MediaConnect).

Output delivery

Delivery method

- Public
Set up the channel to deliver output via the public internet.
- VPC
Set up the channel to deliver output via your Amazon VPC.

You will then be prompted to select two subnets in different availability zones, as well as an existing Security Group

VPC settings

- Select subnets and security groups
- Enter subnet and security group IDs

i Subnets and security groups must be in the the same VPC. The current VPC is vpc-0248b4fdc0cf0a766.

Subnets

Select two subnets in two different Availability Zones

Choose subnets ▼

subnet-0d69f1aff7beef478 ✕
us-east-1d 172.31.0.0/20 vpc-0248b4fdc0cf0a766

subnet-0d2a961d111475999 ✕
us-east-1f 172.31.64.0/20 vpc-0248b4fdc0cf0a766

Security groups

Select security groups

Choose security groups ▼

sg-0c9440db38407309e ✕
launch-wizard-2 created 2021-11-08T18:07:28.666-10:00
vpc-0248b4fdc0cf0a766 Primary SG

2. Create an Input Attachment

Input Attachment network mode should be **Public**, as we are streaming from OBS on a local machine into AWS (as opposed to from another AWS service).

Network mode

- Public**
Specify whitelist rules with CIDR blocks
- VPC**
Use Amazon VPC to specify subnets

Input Security Group is separate and distinct from any existing security groups in [EC2](#) (Elastic Compute Cloud) or VPC. You should allow traffic from wherever you are running OBS (in our example we used

Input security group

Choose an input security group to use with your RTP or RTMP PUSH input type.

- Use existing**
Attach an existing input security group to your channel.
- Create**
Attach a new input security group to your channel.

Use existing

Choose an existing input security group.

Input Security Group 1 1 CIDR block 0.0.0.0/0	3326919 ▼
---	-----------

0.0.0.0/0 which allows traffic from everywhere but offers little security).

Input Destinations is where you specify a name for your **Application Name** and **Instance**. This information will be used to configure the Stream Settings in OBS, so please take note of what you enter here.

Create an Output Group, Add an **Output Group** of type **UDP**, as we will stream via RTP into MediaConnect. In the **UDP Destinations** box, you will have to specify the input endpoint IP of the MediaConnect flow, so you will need to set up that component before finishing with MediaLive.

Input destinations

For RTMP PUSH inputs, you must specify two destination application names and instances.

Input class

For a standard channel, you must choose STANDARD_INPUT. For a single-pipeline channel, you can choose SINGLE_INPUT or STANDARD_INPUT.

SINGLE_INPUT ▼

Destination	Application name and instance	
Destination A	EEGTest	mystream
Destination B	<i>Application name</i>	<i>Application instance</i>

3. Navigate to MediaConnect

For **Source**, choose **VPC Source** (as MediaLive is streaming out into the VPC).

Source

Source type [Info](#)

Standard source

Choose this option if your content comes from a non-entitlement based source, such as an on-premises encoder.

VPC source

Choose this option if your content comes from your VPC.

Entitled source

Choose this option if your content comes from an AWS Elemental MediaConnect flow that is owned by another AWS account.



From the **Protocol** drop-down, choose **RTP**

Protocol

RTP ▼

Specify an **Inbound Port**. You will also need to choose a **VPC Interface** since we are using a VPC Source. If you do not have any existing VPC Interfaces, you will need to create one. Once a VPC Interface has been created, you can choose it from the **VPC Interface** drop-down, then click **Create Flow**.

You will be provided with an **Inbound IP Address** – this is the address needed (along with the port and protocol) for the **Output Destination in MediaLive**.

Sources (1) Info								Manage tags	Details	Update	Remove	Add source
Name	ARN	Type	Inbound IP address	Port	Protocol	Source health						
<input type="radio"/> Inbound-from-MediaLive	 arn:aws:mediacon...	VPC Source	172.31.12.86	8000	rtp	 The flow is inactive						

To finish setting up your MediaConnect Flow, you need to add an output. Output type should be **VPC Output** as we are streaming to Alta hosted on an EC2 in the same VPC. We want to configure Alta for SRT, so for Protocol, choose **SRT Listener**.

Output type

VPC output

Protocol

SRT listener

You will also need to specify a **Port**, because MediaConnect is operating in SRT Listener mode. Finally, in the **Output to VPC** drop-down, choose a **VPC Interface**, then click **Add Output**.

4. **Set up MediaConnect** so it can consume the output from Alta. You can follow the same steps as above to do this, with some exceptions:
 - For **Source Protocol**, choose **SRT Listener** and specify a **Port**
 - For **Output Type**, choose **Standard Output**
 - For **Output Protocol**, choose **SRT Listener** (we will connect with VLC to monitor the output stream in real time), and specify a **Port**
 - After doing this, you will again be provided with an Inbound & Outbound IP address & port. You'll need these for Alta and VLC, respectively

Retrieve the Inbound IP address, port, and protocol information from your MediaConnect Source Settings (from Step 3d above), and in MediaLive, add this to your **MediaLive UDP Destinations**, and then click Update Channel.

Output to MediaConnect

UDP destinations

Choose a destination for your transport streams.

Output reference	Destination A
Output 1	rtp://172.31.12.86:8000

Note: The above are settings highlights for the passing of input and output streams through both MediaLive and MediaConnect. For more detailed info, please see AWS resources [for MediaLive](#) and [for MediaConnect](#).

Navigate to your **Alta instance** and within the channel's **Stream Settings**, for **Primary**, use srt:// and the IP and port of your **Output** from **Step 3 above**. Specifying an external IP with SRT protocol will automatically put Alta in "caller" mode. For **Output**, use srt:// and the IP and port from your **Source Settings (Inbound IP & Port)** from **Step 4 above**.

Stream Settings

Name
Custom name for identifying this instance

Primary
Listen address for the primary video stream

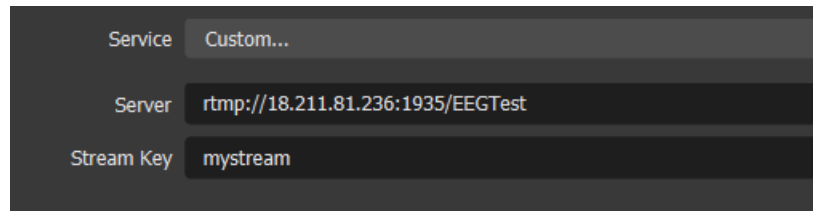
Source
Listen address for a caption source stream used for bridging captions into the primary stream

Output
Destination address for processed primary stream output

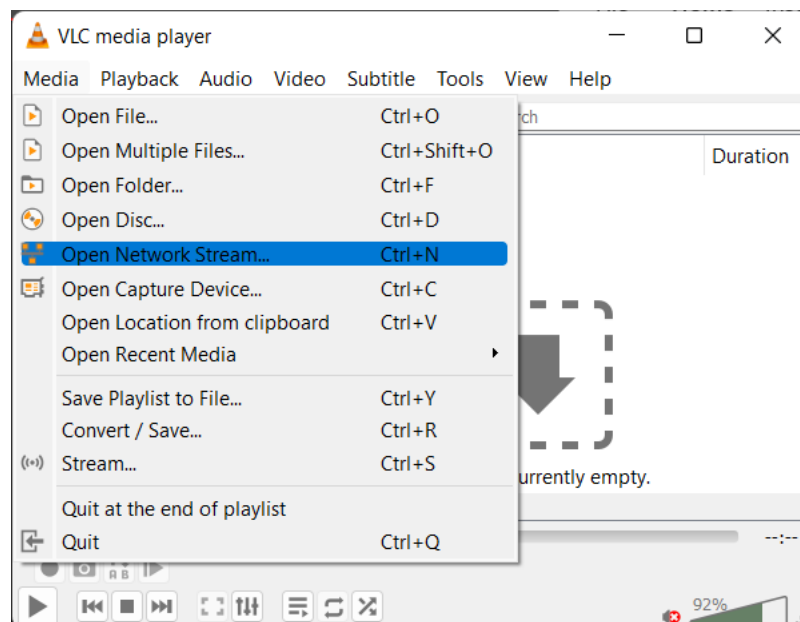
In **iCap Settings**, make sure you fill in the fields for **iCap Company**, **iCap User**, and **iCap Password** with valid credentials to enable the Alta encoder to connect to the iCap network.

In **OBS**, select **Settings**, then **Stream** to access the stream settings

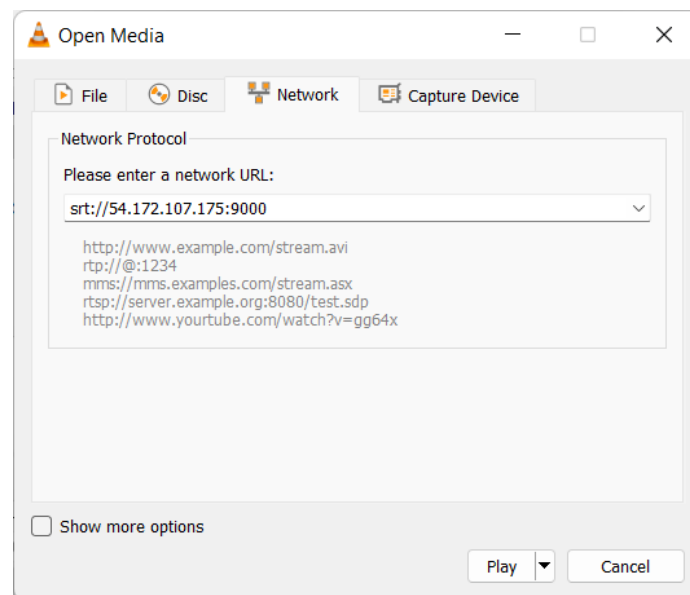
In the **Server** field, enter the **Input Endpoint URL** from your MediaLive Channel's Input Attachment. It should take the form of **rtmp://<ip-address>:1935/<Application Name>**. You created the **Application Name** for the MediaLive Input Destination. In the **Stream Key** field, enter the **Stream Name**, also created for the MediaLive Input Destination. Then click **Apply**.



In **VLC**, click **Media**, then choose **Open Network Stream**.



In the **network URL** field, enter **srt://** followed by the **Output IP address & Port** specified in Step 3 above, and then click **Play**.



5. Start Streaming:

Now that everything is set up, you need to start your MediaLive channel, as well as both your MediaConnect input and output Flows. This process can take a couple of minutes. Once MediaLive is running, you can start streaming from **OBS** by clicking **Start Streaming**. If you click “Play” in VLC to connect to the output stream, it should automatically connect once all the pieces are running and the output stream is detected.

FLOW 2: Using Another Streaming Source (other than OBS)

SRT streams can be created, encoded, and viewed in real time when using any streaming tool/source creator other than OBS (VLC is one example).

The only differences from FLOW 1 above would be the following:

- The input type may differ (VLC does not create RTMP streams, for example, but does create UDP, RTP, http, etc) so the Input Type field selected in MediaLive would be adjusted accordingly.
- The reason you may still send the stream through MediaLive even if not converting from RTMP is because it helps clean up the source stream, since many settings available in OBS are not compatible with professional broadcast standards.

But if desired you can send it directly from the streaming tool to MediaConnect, provided it's an input type supported by MediaConnect and you specify the port.

After creating an input attachment from MediaLive (referenced in step 2 in Flow 1 above), the user should edit their stream settings in whatever they're using as a stream creator (as opposed to within OBS, which was documented in Step 7 of Flow 1 above), to make sure the stream hits the appropriate destination from the source.

If sending directly from the stream source to MediaConnect, you would take the input endpoint provided by MediaConnect and point the stream source to that IP/port when setting up your outbound stream. You also have to make sure the MediaConnect Source is "Public" rather than "VPC", since in that workflow we were going from MediaLive to MediaConnect and they're both in the same VPC. In this case, you'd be trying to get into MC from the public internet.

FLOW 3: Have SRT Source Stream that Can Go Directly into Alta

The simplest flow is if you are able to create a source stream that's SRT without needing to convert it to SRT first via MediaConnect. You also wouldn't send it through MediaLive since it doesn't support SRT as input or output. It can still be sent from iCap Alta to MediaConnect and then to VLC in this case to view the stream live.

In this flow, you would just send it directly from the source into iCap Alta, where it would be encoded. To do this, simply replicate Step 6 from Flow 1 above but enter the IP and Port from the output of your source stream in the **Primary** field.

We hope you found these workflows helpful! iCap Alta provides 24/7 connectivity to captioners, the [Lexi automatic captioning service](#), and [Smart Lexi](#) hybrid human curated and automatic captioning service.

Get in touch to [request a demo of iCap Alta](#).