

## Tvheadend - Feature #5723

### Output UDP stream (for multicast)

2019-09-02 18:17 - Stéphane Duperron

<b>Status:</b>	New	<b>Start date:</b>	2019-09-02
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	John Törnblom	<b>% Done:</b>	0%
<b>Category:</b>	Streaming	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	4.6		
<b>Description</b>			
<p>I am software engineer by Innes company (<a href="http://www.innes.pro">http://www.innes.pro</a>) and we are interested in using Tvheadend as a gateway DVB-T to UDP stream.</p> <p>We are ready to contribute to develop this new feature. I have started writing code for that and tested it successfully on a Raspberry Pi.</p> <p>For now, udp stream starts by getting URL <a href="http://&amp;lt;server_addr&amp;gt;:9981/udpstream/channelnumber/&amp;lt;channelnumber&amp;gt;?address=&amp;lt;udp_dest_addr&amp;gt;&amp;port=&amp;lt;udp_port&amp;gt;">http://&amp;lt;server_addr&amp;gt;:9981/udpstream/channelnumber/&amp;lt;channelnumber&amp;gt;?address=&amp;lt;udp_dest_addr&amp;gt;&amp;port=&amp;lt;udp_port&amp;gt;</a> &amp;lt;udp_dest_addr&amp;gt; where &amp;lt;udp_dest_addr&amp;gt; can be a multicast address.</p> <p>I can share this code as a branch on your github or another way, as you want.</p> <p>We also need that UDP stream starts automatically when Tvheadend starts.</p> <p>Are you interested in implementing (with our help) UDP as output of Tvheadend in the next future ?</p> <p>Thank in advance for your answer</p>			

### History

#### #1 - 2019-09-02 21:19 - Flole Systems

I think it would be better to use the SAT-IP functionality for this, the standard supports multicast, however tvheadend doesn't support this yet as far as I know. You would need to send a RTP request then and start the multicast with that. You will get a RTP stream then.

#### #2 - 2019-09-03 07:55 - saen acro

only solution I found is external software

#### [Tomcast](#)

If someone integrate it to TVH on new tab as "static streaming" will be best.

#### #3 - 2019-09-03 09:17 - Jaroslav Kysela

The multicast streaming can be activated through SAT>IP RTSP commands. An example simple python tool:  
<https://github.com/perexg/satip-axe/blob/master/tools/multicast-rtsp>. We can talk about the channel / service extension instead manual tuning parameters / PID list in the RTSP requests. Also, the SAT>IP RTCP packets might be more controllable.

@Stéphane : Could you show your code through the pull request on github? At least to analyze what's missing in the SAT>IP server code in tvh.

#### #4 - 2019-09-03 14:47 - Stéphane Duperron

Thank you guys for your answers.  
I try Tomcast but have problem with http authentication.

First test (without authentication): Channels configuration file content =  
chan1 239.1.2.3:1234 <http://&lt;tvheadend-ip&gt;:9981/stream/channelnumber/&lt;n>>

In this case tvheadend returns error 401  
-> is it possible to start http streaming from tvheadend without authentication ?

Second test (user/pass in URL): Channels configuration file content =  
chan1 239.1.2.3:1234 <http://&lt;username&gt;:&lt;password&gt;@&lt;tvheadend-ip&gt;:9981/stream/channelnumber/&lt;n>>

In this case tomcast cannot resolve the URL (Sep 03 14:20:07 host unixsol-tomcast: ERR : Can't resolve src host Chan: chan1 ...)  
-> how to provide http credential to Tomcast ?

#### #5 - 2019-09-03 15:17 - saen acro

Create user "" with same password "" and limit it to ip of tomcast

**#6 - 2019-09-03 18:33 - Stéphane Duperron**

Here is my code to have udp streamer inside TVH : <https://github.com/innes-labs/tvheadend/tree/udpstream>  
Waiting for your remark so that it can be integrated in the master branch

**#7 - 2019-09-03 23:36 - Jaroslav Kysela**

It see the main functional issue that you cannot stop UDP streaming with your patch. I would prefer to keep the controlling HTTP connection open and break the UDP output when the HTTP connection closes or just use HTTP to start / stop and run the main streaming task in another thread (which should be probably best - it will allow to add a static multicast table to GUI later without HTTP server misuse).

**#8 - 2019-09-04 09:40 - saen acro**

Suggestion:  
New TAB in UI **Static streaming**

[Configuration | Static streaming | Status | About]  
there need to have toolbar with  
/Add/Remove/Start/Stop/ button's

Add> (new window) with options:

- select mapped service
- set mcas group
- set mcas port
- set mcas interface
- set mcas ttl
- set mcas format (UDP|RTP)
- pass options (EIT|SDT)

in content table  
/ select / play / service name / mcasgroup:mcasport:mcasttl / used adapter+mux or ip stream/ uptime / bitrate graph/ generated traffic /

**#9 - 2019-09-04 09:55 - Stéphane Duperron**

I think also that using HTTP to start/stop the stream and processing it in a specific thread is best.  
I will change the code to do so.

Suggestion for the URL: another parameter 'action' in the query string  
for example

<http://tvheadend/udpstream/channelnumber/1?action=start&address=239.1.2.3&port=1234>  
<http://tvheadend/udpstream/channelnumber/1?action=stop&address=239.1.2.3&port=1234>

**#10 - 2019-09-04 11:59 - saen acro**

What is TTL value and can it be changed?

**#11 - 2019-09-04 14:18 - Stéphane Duperron**

Saen,  
currently TTL has its default (system) value because no setsockopt(IP\_TTL) is done at UDP socket creation.  
This parameter can easily be added in struct udp\_connection and set at socket init

**#12 - 2019-09-05 01:07 - saen acro**

With version is stream?

IGMPv1 = RFC 1112  
IGMPv2 = RFC 2236  
IGMPv3 = RFC 3376  
MLDv1 = RFC 3678

**#13 - 2019-09-06 17:29 - Stéphane Duperron**

Update available on <https://github.com/innes-labs/tvheadend/commit/8bdb7d522aae8ab66444d8cd401f4de20c541346>

As you will see, I finally use following URL to control UDP streams:

<http://tvheadend/udpstream/start/channelnumber/1?address=239.1.2.3&port=1234>  
<http://tvheadend/udpstream/stop/channelnumber/1?address=239.1.2.3&port=1234>

Next step for me is to support parameters proposed by Saen (TTL, IGMP version, interface)  
Is there anybody to take in charge the GUI ? Because I'm not very skilled in this part

**#14 - 2019-09-07 10:36 - saen acro**

+1 Nice job.  
Now [Jaroslav Kysela](#) need to approve commit.

**#15 - 2019-09-16 11:37 - Jaroslav Kysela**

But my main objection is that you don't run the UDP streaming in the different thread. You're misusing the HTTP thread (created for the start request parsing) so you cannot start or use the UDP streaming from the other place in tvh programmatically in the future.

Look for `src/upnp.c - upnp_thread()` for an example. One thread per one UDP output stream. The `http_stream_run()` should be called from the new thread. Also, this function should be renamed and moved to another place with the `udp_stream_t` management like `src/socketstream.c` or so..

**#16 - 2019-09-19 11:56 - Stéphane Duperron**

We are ready to improve UDP code as you suggest. But we need assurance that, once approved, our commit will be merged in master branch and UDP output will become a standard feature of TVH.

When can the feature be planned in TVH roadmap according to you ?

As I said in my previous post, is there anyone to work on the GUI part (New TAB for UDP streaming, socket option's setting) ?

**#17 - 2019-09-19 12:31 - saen acro**

[Stéphane Duperron](#) is your repo branch `udpoutput` have latest patches?

**#18 - 2019-09-19 13:21 - Jaroslav Kysela**

- Target version set to 4.4

**#19 - 2019-09-19 13:27 - Jaroslav Kysela**

Stéphane Duperron wrote:

We are ready to improve UDP code as you suggest. But we need assurance that, once approved, our commit will be merged in master branch and UDP output will become a standard feature of TVH.

If I find your code sufficient and universal for the job, I can merge it. I just don't like the HTTP server misuse (and believe me - you are not first who proposed this feature in this wrong way - in my eyes).

When can the feature be planned in TVH roadmap according to you ?

As I said in my previous post, is there anyone to work on the GUI part (New TAB for UDP streaming, socket option's setting) ?

We can add the GUI part later - you might start with the simple HTTP control as you proposed initially.

**#20 - 2019-09-19 13:48 - saen acro**

[Jaroslav Kysela](#) To be clear

Need to be realised with independent library "udpout.c udpout.h"

Be more specific, fortuneteller is on vacation 

**#21 - 2019-09-19 18:33 - Mark Clarkstone**

I'm just going to stick my nose in here & add my thoughts. 

This is a great feature & something I will be using.

- Please don't tie it to the http server or limit it just to channel streams (reasons to follow).

- Would be awesome to have it blast EPG & maybe even alerts for dvr rules.

- I'd love to see UDP used as another API entry/control point, a rather simple one at that. HTSP is great but it would be nice to have something a lot nicer than JSON, [TOML](#) for example.

**#22 - 2019-10-04 11:49 - Stéphane Duperron**

I have just pushed a new version of UDP streaming (now running in its own thread):

<https://github.com/innes-labs/tvheadend/commit/0e3b9adf527ac2ec1b424c6b4fe93210c2055fbc>

**#23 - 2019-10-18 10:41 - Stéphane Duperron**

Has anyone taken a look at my last commit ? Is it sufficient to be merged ?

**#24 - 2019-10-18 17:10 - saen acro**

[Stéphane Duperron](#) is it possible to stream mux/transponder in same way?

**#25 - 2019-10-18 17:22 - Stéphane Duperron**

saen acro Not yet, just channels and services. But it will be easy to implement (using `http_stream_mux()` as model)

**#26 - 2019-10-18 18:18 - Jaroslav Kysela**

Create pull request, so I can review all changes together... I just commented the udpstream code (move it to another files).

**#27 - 2020-07-18 22:33 - saen acro**

I hope there is moovement on this.

**#28 - 2021-02-20 22:32 - Flole Systems**

- *Target version changed from 4.4 to 4.6*