

Tvheadend - Bug #5496

Only after a reboot all sat>ip tuners are accessible again

2019-01-14 20:02 - Thilo Gebers

Status:	New	Start date:	2019-01-14
Priority:	Normal	Due date:	
Assignee:	Jaroslav Kysela	% Done:	0%
Category:	SAT>IP	Estimated time:	0.00 hour
Target version:		Affected Versions:	3.4, 4.0
Found in version:	4.3		

Description

Currently I'm using tvh in version 4.3-1718~g8e0dd2b.
My SAT>IP Server is a Kathrein EXIP 414.

First of all the problem I will describe did not appear in this version the first time, it has been around for quite some time.

A couple of days after I restart tvh it "lost" the tuners. Tuner #1 and sometimes tuner #2 doesn't work anymore. It doesn't matter if the idle scan starts or if a client tries to get a tv stream. Tuner #1 and tuner #2 are dead. As you can see below the idle scan on tuner #1 ends up in a "no data received" / "scan no data, failed" and the same procedure starts 10 seconds later over and over again.

```
2019-01-14 18:10:29.854 mpegts: 12480V in DVB-S Network - tuning on SAT>IP DVB-S Tuner #1 (10.0.15.1@UDP)
2019-01-14 18:10:29.854 subscription: 7E95: "scan" subscribing to mux "12480V", weight: 2, adapter : "SAT>IP DVB-S Tuner #1 (10.0.15.1@UDP)", network: "DVB-S Network", service: "Raw PID Subscription"
2019-01-14 18:10:35.035 satip: SAT>IP DVB-S Tuner #1 (10.0.15.1@UDP) - no data received, restarting RTSP
2019-01-14 18:10:39.854 mpegts: 12480V in DVB-S Network - scan no data, failed
2019-01-14 18:10:39.854 subscription: 7E95: "scan" unsubscribing
2019-01-14 18:10:49.854 mpegts: 12480V in DVB-S Network - tuning on SAT>IP DVB-S Tuner #1 (10.0.15.1@UDP)
2019-01-14 18:10:49.854 subscription: 7E97: "scan" subscribing to mux "12480V", weight: 2, adapter : "SAT>IP DVB-S Tuner #1 (10.0.15.1@UDP)", network: "DVB-S Network", service: "Raw PID Subscription"
2019-01-14 18:10:55.005 satip: SAT>IP DVB-S Tuner #1 (10.0.15.1@UDP) - no data received, restarting RTSP
2019-01-14 18:10:59.854 mpegts: 12480V in DVB-S Network - scan no data, failed
2019-01-14 18:10:59.854 subscription: 7E97: "scan" unsubscribing
```

The same issue on tuner #1 and tuner #2 with a client trying to stream a tv channel but it only works on tuner #4.
(It skips tuner #3 because it's currently in use by another client)

```
2019-01-14 18:58:14.154 mpegts: 12421.5H in DVB-S Network - tuning on SAT>IP DVB-S Tuner #1 (10.0.15.1@UDP)
2019-01-14 18:58:14.154 subscription: 7FB4: "HTTP" subscribing on channel "WDR HD Essen", weight: 100, adapter: "SAT>IP DVB-S Tuner #1 (10.0.15.1@UDP)", network: "DVB-S Network", mux: "12421.5H", provider: "ARD", service: "WDR HD Essen", profile="pass", hostname="10.0.13.1", client="VLC/3.0.4 LibVLC/3.0.4"
2019-01-14 18:58:19.398 satip: SAT>IP DVB-S Tuner #1 (10.0.15.1@UDP) - no data received, restarting RTSP
2019-01-14 18:58:20.155 subscription: 7FB4: service instance is bad, reason: Tuning failed
2019-01-14 18:58:20.155 mpegts: 12421.5H in DVB-S Network - tuning on SAT>IP DVB-S Tuner #2 (10.0.15.1@UDP)
2019-01-14 18:58:20.155 subscription: 7FB4: "HTTP" subscribing on channel "WDR HD Essen", weight: 100, adapter: "SAT>IP DVB-S Tuner #2 (10.0.15.1@UDP)", network: "DVB-S Network", mux: "12421.5H", provider: "ARD", service: "WDR HD Essen", profile="pass", hostname="10.0.13.1", client="VLC/3.0.4 LibVLC/3.0.4"
2019-01-14 18:58:30.155 mpegts: 12421.5H in DVB-S Network - scan no data, failed
2019-01-14 18:58:30.155 subscription: 7FB4: service instance is bad, reason: No input detected
2019-01-14 18:58:30.155 mpegts: 12421.5H in DVB-S Network - tuning on SAT>IP DVB-S Tuner #4 (10.0.15.1@UDP)
2019-01-14 18:58:30.155 subscription: 7FB4: "HTTP" subscribing on channel "WDR HD Essen", weight:
```

```
100, adapter: "SAT>IP DVB-S Tuner #4 (10.0.15.1@UDP)", network: "DVB-S Network", mux: "12421.5H",
provider: "ARD", service: "WDR HD Essen", profile="pass", hostname="10.0.13.1", client="VLC/3.0.4
LibVLC/3.0.4"
```

```
2019-01-14 18:58:34.775 mpegts: 12421.5H in DVB-S Network scan complete
```

If I restart tvh everything works as expected.

```
2019-01-14 19:01:19.085 mpegts: 11361.75H in DVB-S Network - tuning on SAT>IP DVB-S Tuner #1 (10.0
.15.1@UDP)
```

```
2019-01-14 19:01:19.085 subscription: 7C62: "10.0.11.39 [ Kodi Media Center ]" subscribing on chan
nel "ZDF HD", weight: 100, adapter: "SAT>IP DVB-S Tuner #1 (10.0.15.1@UDP)", network: "DVB-S Netwo
rk", mux: "11361.75H", provider: "ZDFvision", service: "ZDF HD", profile="htsp", hostname="10.0.11
.39", username="10.0.11.39", client="Kodi Media Center"
```

```
2019-01-14 19:04:02.592 mpegts: 12421.5H in DVB-S Network - tuning on SAT>IP DVB-S Tuner #2 (10.0.
15.1@UDP)
```

```
2019-01-14 19:04:02.592 subscription: 7FCD: "HTTP" subscribing on channel "WDR HD Essen", weight:
100, adapter: "SAT>IP DVB-S Tuner #2 (10.0.15.1@UDP)", network: "DVB-S Network", mux: "12421.5H",
provider: "ARD", service: "WDR HD Essen", profile="pass", hostname="10.0.13.1", client="VLC/3.0.4
LibVLC/3.0.4"
```

```
2019-01-14 19:04:03.842 mpegts: 12421.5H in DVB-S Network scan complete
```

History

#1 - 2019-01-15 08:58 - Jaroslav Kysela

Try to play with 'Fast input switch' and forced UDP RTP port settings. The best debugging method is to capture the network communication, if the satip server does the right job.

#2 - 2019-01-15 12:25 - Thilo Gebers

Jaroslav Kysela wrote:

Try to play with 'Fast input switch' and forced UDP RTP port settings.

'Fast input switch' was already activated.

I've assigned an UDP port to each tuner, in increments of ten, tuner #1 50010, tuner #2 50020, and so on.

Now I will leave tvh untouched for a week. No setting adjustment, no restart. After that, I'll give you feedback whether the port customizations have resolved the issues.

#3 - 2019-01-15 12:30 - Jaroslav Kysela

Thilo Gebers wrote:

Jaroslav Kysela wrote:

Try to play with 'Fast input switch' and forced UDP RTP port settings.

'Fast input switch' was already activated.

It's the default. Try to turn this off, too.

#4 - 2019-01-15 12:36 - Thilo Gebers

Jaroslav Kysela wrote:

Thilo Gebers wrote:

Jaroslav Kysela wrote:

Try to play with 'Fast input switch' and forced UDP RTP port settings.

'Fast input switch' was already activated.

It's the default. Try to turn this off, too.

All right ... It is turned off now.

#5 - 2019-01-24 14:57 - Thilo Gebers

For over a week, Tvheadend runs perfectly with the two changed settings. The two tuners are still found and used by Tvheadend. Many thanks to Jaroslav for his support.
In order to find out which setting was responsible for the problems, I will reactivate "Fast input switch" and observe how this affects tuner #1 and #2.

#6 - 2019-03-01 10:12 - Thilo Gebers

After a good month, TvHeadend is very stable. All tuners are still available. In the meantime version 4.3-1768 is in use. Even so, there are no problems.
I set the following settings: an UDP port was assigned to each tuner. I deactivated "Fast input switch". In addition, I have disabled the "idle scan" on all tuners.

Now we can set the topic to "solved".

Jaroslav, thanks again for your support!

#7 - 2019-09-18 16:49 - Flole Systems

I'm having the exact same issue, even without "Fast input switch" and a static UDP Port assigned. Tvheadend does in this case not receive any UDP Packets (while wireshark shows that they are there). There's also no "ICMP Port unreachable" until shortly after the teardown (I think this should not happen, we should first wait for the response of the server and then close the port), so I assume the port is indeed open the entire time.

Also I think that the last Tuner Status (SNR and Signal Level) is not cleared, if the tuner doesn't send those information the last one will stay in there. Maybe there is something wrong somewhere with the (de)initialization?

@Jaroslav: Any assumption what might be going on here? I'm more than happy to dive into the debugging but I have no idea what could cause a behaviour like this.

#8 - 2019-09-18 17:34 - Flole Systems

Actually after waiting a little they seem to start working again.....

#9 - 2019-09-18 23:24 - Pablo R.

The same happens here, I think I've solved it by increasing the kernel udp buffer. Although I am still not clear at all.
What I have noticed is that the adapters are hung when many frequencies are closed together, for example closing 9 at the same time. Some of them remains at 0 until tvheadend restart.

#10 - 2019-09-21 00:45 - Flole Systems

Today I noticed that sometimes tvheadend doesnt even do the RTP connection to start the Stream. I will need to do some GDB Debugging but first I need to figure out the points that I want to stop and look at (probably right before the RTP is attempting to connect and right before the UDP Socket is prepared and opened).

#11 - 2019-09-23 19:53 - Flole Systems

I've done a little more investigation now: Usually we should have satip_frontend.c:2081 receiving the UDP Data, for some reason this is not happening though, the code never reaches 2081 when the tuner is in this unusable state. The problem must be somewhere in the lines above....

@Jaroslav: Is this thread started for every new tuning or is it setup once when the input source is setup and then re-used? Any idea what I should look for when trying to find the cause of the issue?

#12 - 2019-09-24 07:52 - saen acro

[Flole Systems](#) graphviz show connection flow and some dead code

#13 - 2019-09-24 09:40 - Jaroslav Kysela

Flole Systems wrote:

I've done a little more investigation now: Usually we should have satip_frontend.c:2081 receiving the UDP Data, for some reason this is not happening though, the code never reaches 2081 when the tuner is in this unusable state. The problem must be somewhere in the lines above....

@Jaroslav: Is this thread started for every new tuning or is it setup once when the input source is setup and then re-used? Any idea what I should look for when trying to find the cause of the issue?

It's reused for the fast switch. You may try to analyze '--trace satip' log and eventually add more traces to see where the code gets stuck.

#14 - 2019-10-17 17:10 - Flole Systems

I've added additional traces and I am seeing:

```
satip: Falling through play  
satip: ev0.ptr != rtp, returning!
```

So we properly execute until `satip_frontend.c:2079` (I have added another line after that which indicates if we are ready to receive which doesn't get executed when the issue occurs) but then something bad happens on the next run in the loop. I have checked `exit_flag` as well as `fatal` and those are not 0 at that point, so we are not exiting early. Unfortunately for me it now fixed itself again so I have to wait for it to happen again. I suspect this is due to the `satip` server not responding or not responding properly to one request and something in `tvheadend` doesn't get cleaned up properly then affecting any other following request (for example a non-closed port which would cause any port open operation on the same port to fail for some time until the system automatically frees the port).

#15 - 2019-10-22 23:43 - Pablo R.

Flole Systems, Have you managed to detect the culprit of this problem?

Since I put the UDP ports as static and disabled Fast input I have not had problems again, but it would be nice if the bug could be fixed.

#16 - 2019-10-23 00:42 - Flole Systems

I'm trying hard to get this fixed before the next big release but unfortunately I need to wait for the issue to occur in order to debug what's going on, so I need to be at my computer, have time and notice that a tuner is not working, you can probably imagine how often that's the case `wink.png`

#17 - 2019-10-23 08:27 - Jaroslav Kysela

Perhaps, it may be helpful to attach `gdb` when this occurs and look for "tvh:satip-front" threads to look, where things break up.

#18 - 2019-10-23 08:35 - saen acro

Strange solution without reboot.
Disable/enable or viceversa `sat>ip` server in `webui`

Guessing but time to get response is too short.

#19 - 2019-10-23 13:12 - Flole Systems

I've tried `gdb`, the problem is that `tvheadend` tries to go through all tuners then giving me thousands of threads in `gdb` making it super hard to figure out which one is the right one. Also I'm always hitting the timeout when using `gdb`. I've added debug output to everything that could possibly lockup so I will hopefully be able to see it next time.

#20 - 2019-10-25 15:26 - Pablo R.

The problem has appeared again, I rectify.

Static ports and remove Fast input do not solve it.

#21 - 2019-10-25 16:18 - Flole Systems

I have more information though: Unfortunately I forgot one place in the loop where I wanted to add my debug output, and seems like that's the spot that's triggering it. The line I suspect is

```
if (nfdns < 1) continue;
```

as the loop is constantly being run but the code is not reaching any other `continue`-instruction (or I forgot another one).

@Jaroslav: Does that already help you somehow? If I had to guess I'd say that maybe a pipe contains data for the old thread and the new thread reads it and then causes issues, or something similar.

#22 - 2019-10-25 19:16 - Jaroslav Kysela

The thread lifetime is similar to the frontend lifetime, thus there should not be a problem with the concurrent threads. It appears like RTSP is fine, but RTP/UDP stalls for a reason.

#23 - 2019-10-25 19:32 - Flole Systems

Yes and no, we never reach line 2094 which takes care of the RTSP Status receiving, I do not know where the RTP traffic receiving should happen, I thought that's in that function somewhere. If not I need to add more debugging output there.

#24 - 2019-10-25 20:11 - Flole Systems

Maybe my previous post is unclear: We properly do the RTSP for setting up the channel and the server is sending to the correct port specified, just the status receiving is never done as well as the RTP receiving (which I thought is somewhere in that function as well but I haven't found it yet).

#25 - 2019-10-25 23:10 - Jaroslav Kysela

Yes, RTP packets should be read in line 2094 (`udp_multirecv_read` call). Apparently, the `tvhpoll_wait()` is not working as expected (perhaps UDP/RTP and UDP/RTCP file descriptors are in a wrong state or so).

#26 - 2019-10-26 01:52 - Flole Systems

What variables should I look at at that point? Tvhpoll_wait is supposed to wait for the event that triggers when the first data arrives? I can probably even use gdb now that I found a point where only the non-working tuner is stopping, so let me know what I should look for next time I experience this.

#27 - 2019-10-26 08:50 - Jaroslav Kysela

The poll is behind the tvhpoll_wait() (*man 3 poll*). You need to check the socket (file descriptor) state.

#28 - 2019-10-26 09:15 - Jaroslav Kysela

You may try to add FIONREAD (SIOCINQ) ioctl for the UDP file descriptors to monitor the state.

#29 - 2019-10-27 02:06 - Flole Systems

So I'll check if rtp->fd is not null (if it is I'll print a big warning) and then just do the ioctl there. For good measure I'll do the same thing for rtcp. I'll add that tomorrow and then wait for the next lockup. If anybody has another idea what I should add please kindly make suggestions.

Also I noticed that the Webinterface shows old satip tuner signal status after stopping, waiting and starting a stream until there is a status message, not sure if that's a client (browser) or server thing though, if the client caches this it's no problem (or at least not related), if the server does send that as current status it could indicate another incomplete cleanup somewhere.

#30 - 2019-10-27 23:48 - Pablo R.

[Flole Systems](#) Systems If you provide me with a patch with the necessary TRACES, I could check them when the problem appears for me.

#31 - 2019-10-27 23:51 - Flole Systems

More info:

```
Oct 27 23:44:31 tvheadend[40289]: satip: No NFDS, Continuing!
Oct 27 23:44:31 tvheadend[40289]: satip: RTCP Bytes available: 0
Oct 27 23:44:31 tvheadend[40289]: satip: RTP Bytes available: 0
Oct 27 23:44:31 tvheadend[40289]: satip: Looping
Oct 27 23:44:31 tvheadend[40289]: satip: (ev[0].ptr == rtcp)
Oct 27 23:44:31 tvheadend[40289]: satip: Looping
Oct 27 23:44:31 tvheadend[40289]: satip: Continuing after change!
Oct 27 23:44:31 tvheadend[40289]: satip: Looping
Oct 27 23:44:32 tvheadend[40289]: satip: No NFDS, Continuing!
Oct 27 23:44:32 tvheadend[40289]: satip: RTCP Bytes available: 80
Oct 27 23:44:32 tvheadend[40289]: satip: RTP Bytes available: 1328
Oct 27 23:44:32 tvheadend[40289]: satip: Looping
Oct 27 23:44:32 tvheadend[40289]: satip: No NFDS, Continuing!
Oct 27 23:44:32 tvheadend[40289]: satip: RTCP Bytes available: 80
Oct 27 23:44:32 tvheadend[40289]: satip: RTP Bytes available: 1328
Oct 27 23:44:32 tvheadend[40289]: satip: Looping
Oct 27 23:44:33 tvheadend[40289]: satip: No NFDS, Continuing!
Oct 27 23:44:33 tvheadend[40289]: satip: RTCP Bytes available: 80
Oct 27 23:44:33 tvheadend[40289]: satip: RTP Bytes available: 1328
Oct 27 23:44:33 tvheadend[40289]: satip: Looping
Oct 27 23:44:33 tvheadend[40289]: satip: No NFDS, Continuing!
Oct 27 23:44:33 tvheadend[40289]: satip: RTCP Bytes available: 80
Oct 27 23:44:33 tvheadend[40289]: satip: RTP Bytes available: 1328
Oct 27 23:44:33 tvheadend[40289]: satip: Looping
Oct 27 23:44:34 tvheadend[40289]: satip: No NFDS, Continuing!
Oct 27 23:44:34 tvheadend[40289]: satip: RTCP Bytes available: 80
Oct 27 23:44:34 tvheadend[40289]: satip: RTP Bytes available: 1328
Oct 27 23:44:34 tvheadend[40289]: satip: Looping
Oct 27 23:44:34 tvheadend[40289]: satip: No NFDS, Continuing!
Oct 27 23:44:34 tvheadend[40289]: satip: RTCP Bytes available: 80
Oct 27 23:44:34 tvheadend[40289]: satip: RTP Bytes available: 1328
Oct 27 23:44:34 tvheadend[40289]: satip: Looping
Oct 27 23:44:35 tvheadend[40289]: subscription: 0019: service instance is bad, reason: No input detected
```

"Looping" is at the satip_frontend.c:1914, the "No nfd..." and following 2 lines are from 1962, "ev⁰.ptr == rtcp" is from 2080. So it seems like there is data available at that point but the event is not properly notifying the thread about that?

This happened after switching channels fast (zapping through them).

#32 - 2019-10-28 13:58 - Jaroslav Kysela

It seems that there is something wrong with the poll. I will try to prepare tvhpoll_dump() or something. Perhaps, we can add tracing to the poll functions, too.

#33 - 2019-10-28 18:23 - Jaroslav Kysela

I added tvhpoll_set_trace() function to tvhpoll.c - commit 4eac68f52a132de8313f2c1fcdcc227df540b2b2 . Perhaps, it may help you to find the issue.

Activate those traces only for the problematic frontend.

#34 - 2019-10-28 20:00 - Flole Systems

Just for documentation:

Setting this up (for SAT-IP) is as easy as adding after satip_frontend.c:1603

```
tvhpoll_set_trace(efd, LS_SATIP, 1);
```

#35 - 2019-10-29 16:25 - Flole Systems

After zapping through the channels I had a different problem this time: The SAT->IP Server never got the request from tvheadend, which was having some weird issues:

```
Oct 29 16:16:29 tvheadend[41868]: satip: SAT>IP DVB-C Tuner #1 (x.x.x.x@UDP) - starting XXXXX in XXXXX
Oct 29 16:16:29 tvheadend[41868]: satip: epoll wait: events=1 ptr=(nil)
Oct 29 16:16:29 tvheadend[41868]: satip: SAT>IP DVB-C Tuner #1 (x.x.x.x@UDP) - start
Oct 29 16:16:29 tvheadend[41868]: capmt: TVHeadend: Starting CAPMT server for service "XXXXXX" on adapter 0
Oct 29 16:16:29 tvheadend[41868]: subscription: 0277: "x.x.x.x [ XXXXX | Kodi Media Center ]" subscribing on c
hannel "XXXX", weight: 150, adapter: "SAT>IP DVB-C Tuner #1 (x.x.x.x@UDP)", network: "XXXXX", mux: "XXXXX", pr
ovider: "XXXX", service: "XXXXX", profile="htsp", hostname="x.x.x.x", username="XXXXXX", client="Kodi Media Ce
nter"
Oct 29 16:16:29 tvheadend[41868]: satip: SAT>IP DVB-C Tuner - local RTP port 49340 RTCP port 49341
Oct 29 16:16:29 tvheadend[41868]: satip: 00CB: SETUP params - &freq=102&sr=6900&msys=dvbc&mtype=256qam&fec=35&
specinv=1&pids=0
Oct 29 16:16:29 tvheadend[41868]: satip: epoll mod: fd=111 events=3 oevents=1 ptr=0x5630ead92f40
Oct 29 16:16:29 tvheadend[41868]: tvhpoll: epoll mod failed [No such file or directory]
Oct 29 16:16:29 tvheadend[41868]: satip: epoll mod: fd=111 events=3 oevents=1 ptr=0x5630ead92f40
Oct 29 16:16:29 tvheadend[41868]: tvhpoll: epoll mod failed [No such file or directory]
```

This is not really related to the issue I had before where data was received but also it's not the first time I am seeing this. Should I open a separate issue for this so we can distinguish between "Tvheadend not receiving UDP traffic" and "Tvheadend not subscribing properly"?

#36 - 2019-10-29 16:45 - Flole Systems

After a restart and doing the same thing, zapping through the channels until I see that tuner 1 is no longer getting data, giving it some "cool down time" (1 to 2 minutes) and then using another client to do the logs, the original issue also appeared:

```
Oct 29 16:36:43 tvheadend[32125]: mpegts: 146MHz in XXXXXXXX - tuning on SAT>IP DVB-C Tuner #1 (XXXXXXX@UDP)
Oct 29 16:36:43 tvheadend[32125]: satip: SAT>IP DVB-C Tuner #1 (XXXXXXX@UDP) - starting 146MHz in XXXXXXXX
Oct 29 16:36:43 tvheadend[32125]: satip: epoll wait: events=1 ptr=(nil)
Oct 29 16:36:43 tvheadend[32125]: satip: SAT>IP DVB-C Tuner #1 (XXXXXXX@UDP) - start
Oct 29 16:36:43 tvheadend[32125]: capmt: TVHeadend: Starting CAPMT server for service "XXXXXXX" on adapter 0
Oct 29 16:36:43 tvheadend[32125]: satip: SAT>IP DVB-C Tuner - local RTP port 36306 RTCP port 36307
Oct 29 16:36:43 tvheadend[32125]: subscription: 016A: "XXXXXXX [ XXXXXXXX | Kodi Media Center ]" subscribing on
channel "XXXXXXX", weight: 150, adapter: "SAT>IP DVB-C Tuner #1 (XXXXXXX@UDP)$
Oct 29 16:36:43 tvheadend[32125]: satip: epoll add: fd=96 events=1 ptr=0x563c218af6e0
Oct 29 16:36:43 tvheadend[32125]: satip: epoll add: fd=97 events=1 ptr=0x563c1f831010
Oct 29 16:36:43 tvheadend[32125]: satip: 0069: SETUP params - &freq=146&sr=6900&msys=dvbc&mtype=256qam&fec=35&
specinv=1&pids=0
Oct 29 16:36:43 tvheadend[32125]: satip: epoll mod: fd=97 events=3 oevents=1 ptr=0x563c1f831010
Oct 29 16:36:43 tvheadend[32125]: satip: Looping
Oct 29 16:36:43 tvheadend[32125]: satip: epoll wait: events=1 ptr=(nil)
Oct 29 16:36:43 tvheadend[32125]: satip: Reading as we have 1
Oct 29 16:36:43 tvheadend[32125]: satip: Read 1
Oct 29 16:36:43 tvheadend[32125]: satip: Looping
Oct 29 16:36:43 tvheadend[32125]: satip: epoll wait: events=2 ptr=0x563c1f831010
Oct 29 16:36:43 tvheadend[32125]: satip: epoll mod: fd=97 events=1 oevents=3 ptr=0x563c1f831010
Oct 29 16:36:43 tvheadend[32125]: satip: (ev[0].ptr == rtcp)
Oct 29 16:36:43 tvheadend[32125]: satip: Looping
Oct 29 16:36:43 tvheadend[32125]: satip: epoll wait: events=1 ptr=0x563c1f831010
Oct 29 16:36:43 tvheadend[32125]: satip: XXXXXXXX #1 - new session 482828139 stream id 1281
Oct 29 16:36:43 tvheadend[32125]: satip: 0069: PLAY params[0] - addpids=104,6640,6641,6642,6644,6646
Oct 29 16:36:43 tvheadend[32125]: satip: PID Changed, continuing!
Oct 29 16:36:43 tvheadend[32125]: satip: Looping
Oct 29 16:36:43 tvheadend[32125]: satip: No NFDS, Continuing!
Oct 29 16:36:43 tvheadend[32125]: satip: RTCP Bytes available: 0
Oct 29 16:36:43 tvheadend[32125]: satip: RTP Bytes available: 0
Oct 29 16:36:43 tvheadend[32125]: satip: Looping
Oct 29 16:36:43 tvheadend[32125]: satip: No NFDS, Continuing!
Oct 29 16:36:43 tvheadend[32125]: satip: RTCP Bytes available: 0
Oct 29 16:36:43 tvheadend[32125]: satip: RTP Bytes available: 0
Oct 29 16:36:43 tvheadend[32125]: satip: Looping
```

```

Oct 29 16:36:43 tvheadend[32125]: satip: epoll wait: events=1 ptr=(nil)
Oct 29 16:36:43 tvheadend[32125]: satip: Reading as we have 1
Oct 29 16:36:43 tvheadend[32125]: satip: Read 1
Oct 29 16:36:43 tvheadend[32125]: satip: Looping
Oct 29 16:36:43 tvheadend[32125]: satip: No NFDS, Continuing!
Oct 29 16:36:43 tvheadend[32125]: satip: RTCP Bytes available: 0
Oct 29 16:36:43 tvheadend[32125]: satip: RTP Bytes available: 0
Oct 29 16:36:43 tvheadend[32125]: satip: Looping
....
Oct 29 16:36:44 tvheadend[32125]: satip: Looping
Oct 29 16:36:44 tvheadend[32125]: satip: epoll wait: events=1 ptr=0x563c1f831010
Oct 29 16:36:44 tvheadend[32125]: satip: 0069: PLAY params[0] - addpids=1,16,17
Oct 29 16:36:44 tvheadend[32125]: satip: PID Changed, continuing!
Oct 29 16:36:44 tvheadend[32125]: satip: Looping
Oct 29 16:36:44 tvheadend[32125]: satip: epoll wait: events=1 ptr=0x563c218af6e0
Oct 29 16:36:44 tvheadend[32125]: satip: Locking DVR Lock
Oct 29 16:36:44 tvheadend[32125]: satip: Locked DVR Lock
Oct 29 16:36:44 tvheadend[32125]: satip: Status string: 'ver=1.2;tuner=1281,240,1,15,146.00,,dvbc,qam256,6900,
,,,1;pids='
Oct 29 16:36:44 tvheadend[32125]: satip: (ev[0].ptr == rtcp)
Oct 29 16:36:44 tvheadend[32125]: satip: Looping
Oct 29 16:36:44 tvheadend[32125]: satip: epoll wait: events=1 ptr=0x563c1f831010
Oct 29 16:36:44 tvheadend[32125]: satip: (ev[0].ptr == rtcp)
Oct 29 16:36:44 tvheadend[32125]: satip: Looping
Oct 29 16:36:44 tvheadend[32125]: satip: Continuing after change!
Oct 29 16:36:44 tvheadend[32125]: satip: Looping
Oct 29 16:36:44 tvheadend[32125]: satip: epoll wait: events=1 ptr=0x563c218af6e0
Oct 29 16:36:44 tvheadend[32125]: satip: Locking DVR Lock
Oct 29 16:36:44 tvheadend[32125]: satip: Locked DVR Lock
Oct 29 16:36:44 tvheadend[32125]: satip: Status string: 'ver=1.2;tuner=1281,240,1,15,146.00,,dvbc,qam256,6900,
,,,1;pids=0,104,6640,6641,6642,6644,6646,1,16,17'
Oct 29 16:36:44 tvheadend[32125]: satip: (ev[0].ptr == rtcp)
Oct 29 16:36:44 tvheadend[32125]: satip: Looping
Oct 29 16:36:45 tvheadend[32125]: satip: No NFDS, Continuing!
Oct 29 16:36:45 tvheadend[32125]: satip: RTCP Bytes available: 0
Oct 29 16:36:45 tvheadend[32125]: satip: RTP Bytes available: 1328
Oct 29 16:36:45 tvheadend[32125]: satip: Looping
Oct 29 16:36:45 tvheadend[32125]: satip: epoll wait: events=1 ptr=0x563c218af6e0
Oct 29 16:36:45 tvheadend[32125]: satip: Locking DVR Lock
Oct 29 16:36:45 tvheadend[32125]: satip: Locked DVR Lock
Oct 29 16:36:45 tvheadend[32125]: satip: Status string: 'ver=1.2;tuner=1281,240,1,15,146.00,,dvbc,qam256,6900,
,,,1;pids=0,104,6640,6641,6642,6644,6646,1,16,17'
Oct 29 16:36:45 tvheadend[32125]: satip: (ev[0].ptr == rtcp)
Oct 29 16:36:45 tvheadend[32125]: satip: Looping
Oct 29 16:36:45 tvheadend[32125]: satip: No NFDS, Continuing!
Oct 29 16:36:45 tvheadend[32125]: satip: RTCP Bytes available: 0
Oct 29 16:36:45 tvheadend[32125]: satip: RTP Bytes available: 1328
Oct 29 16:36:45 tvheadend[32125]: satip: Looping
Oct 29 16:36:46 tvheadend[32125]: satip: epoll wait: events=1 ptr=0x563c218af6e0
Oct 29 16:36:46 tvheadend[32125]: satip: Locking DVR Lock
Oct 29 16:36:46 tvheadend[32125]: satip: Locked DVR Lock
Oct 29 16:36:46 tvheadend[32125]: satip: Status string: 'ver=1.2;tuner=1281,240,1,15,146.00,,dvbc,qam256,6900,
,,,1;pids=0,104,6640,6641,6642,6644,6646,1,16,17'
Oct 29 16:36:46 tvheadend[32125]: satip: (ev[0].ptr == rtcp)

```

Is it possible that the data is available before the eventpoll so it wont trigger when more data is available?

#37 - 2019-10-29 18:26 - Jaroslav Kysela

- Affected Versions 2.4 added

The lines bellow look weird. The two UDP sockets should be registered in the start procedure (line 1838), but it seems that fd=97 is the RTSP TCP socket (judging from the 'epoll mod' line):

```

Oct 29 16:36:43 tvheadend[32125]: satip: epoll add: fd=96 events=1 ptr=0x563c218af6e0
Oct 29 16:36:43 tvheadend[32125]: satip: epoll add: fd=97 events=1 ptr=0x563c1f831010
Oct 29 16:36:43 tvheadend[32125]: satip: 0069: SETUP params - &freq=146&sr=6900&msys=dvbc&mtype=256qam&fec=35&
specinv=1&pids=0
Oct 29 16:36:43 tvheadend[32125]: satip: epoll mod: fd=97 events=3 oevents=1 ptr=0x563c1f831010

```

Flole Systems wrote:

After zapping through the channels I had a different problem this time: The SAT->IP Server never got the request from tvheadend, which was having some weird issues:

[...]

This is not really related to the issue I had before where data was received but also it's not the first time I am seeing this. Should I open a separate issue for this so we can distinguish between "Tvheadend not receiving UDP traffic" and "Tvheadend not subscribing properly"?

It's ok to debug this here, too.

#38 - 2019-10-29 18:27 - Jaroslav Kysela

- Affected Versions 4.4 added

- Affected Versions deleted (2.4)

#39 - 2019-10-31 14:11 - Flole Systems

Seems like something is indeed going missing there. Modified Code:

```
memset(ev, 0, sizeof(ev));
nfds = 0;
if ((rtsp_flags & SATIP_SETUP_TCP) == 0) {
    tvhtrace(LS_SATIP, "Setting up RTP event for %d at %d", rtp->fd, nfds);
    ev[nfds].events = TVHPOLL_IN;
    ev[nfds].fd     = rtp->fd;
    ev[nfds].ptr    = rtp;
    nfds++;
    tvhtrace(LS_SATIP, "Setting up RTCP event for %d at %d", rtcp->fd, nfds);
    ev[nfds].events = TVHPOLL_IN;
    ev[nfds].fd     = rtcp->fd;
    ev[nfds].ptr    = rtcp;
    nfds++;
} else {
    tvhtrace(LS_SATIP, "Not setting up events as we are doing TCP");
    rtsp->hc_io_size      = 128 * 1024;
    rtsp->hc_rtp_data_received = satip_frontend_rtp_data_received;
}
if (i) {
    tvhtrace(LS_SATIP, "Setting up RTSP event for %d at %d", rtsp->hc_fd, nfds);
    ev[nfds].events = TVHPOLL_IN;
    ev[nfds].fd     = rtsp->hc_fd;
    ev[nfds].ptr    = rtsp;
    nfds++;
}
tvhtrace(LS_SATIP, "Actually adding our list");
tvhpoll_add(efd, ev, nfds);
rtsp->hc_efd = efd;
```

Logs:

```
Oct 31 14:08:32 tvheadend[36949]: satip: SAT>IP DVB-C Tuner - local RTP port 59788 RTCP port 59789
Oct 31 14:08:32 tvheadend[36949]: satip: Setting up RTP event for 116 at 0
Oct 31 14:08:32 tvheadend[36949]: satip: Setting up RTCP event for 117 at 1
Oct 31 14:08:32 tvheadend[36949]: satip: Setting up RTSP event for 119 at 2
Oct 31 14:08:32 tvheadend[36949]: satip: Actually adding our list
Oct 31 14:08:32 tvheadend[36949]: satip: epoll add: fd=117 events=1 ptr=0x558184cc60b0
Oct 31 14:08:32 tvheadend[36949]: satip: epoll add: fd=119 events=1 ptr=0x55818394be00
Oct 31 14:08:32 tvheadend[36949]: satip: 0050: SETUP params - XXX
Oct 31 14:08:32 tvheadend[36949]: satip: epoll mod: fd=119 events=3 oevents=1 ptr=0x55818394be00
```

Any idea what could be going on here? Would GDB help in this case?

#40 - 2019-10-31 15:19 - Jaroslav Kysela

- Affected Versions 3.4, 4.0 added

- Affected Versions deleted (4.4)

Do you see 'epoll rem' lines before the new problematic tune? Could you check if RTP fd there? The RTP/RTCP fds should be removed at line 2131 (tvhpoll_rem call).

#41 - 2019-10-31 20:06 - Flole Systems

Shouldnt a new thread start for this anyways as I have "Fast Input Switch" disabled and all the old variables should be "forgotten"/reinitialized?

I looked at the stopping part of the tune shown above and there's this:

```
Oct 31 14:08:37 tvheadend[36949]: subscription: 0090: service instance is bad, reason: No input detected
Oct 31 14:08:37 tvheadend[36949]: satip: SAT>IP DVB-C Tuner #1 (XXXX@UDP) - stopping XXXMHz in XXX
Oct 31 14:08:37 tvheadend[36949]: satip: epoll wait: events=1 ptr=(nil)
Oct 31 14:08:37 tvheadend[36949]: satip: Reading as we have 1
Oct 31 14:08:37 tvheadend[36949]: satip: Read 1
Oct 31 14:08:37 tvheadend[36949]: satip: SAT>IP DVB-C Tuner #1 (XXXX@UDP) - input thread received mux close
Oct 31 14:08:37 tvheadend[36949]: satip: epoll rem: fd=116 events=1
Oct 31 14:08:37 tvheadend[36949]: tvhpoll: epoll del failed [No such file or directory]
Oct 31 14:08:37 tvheadend[36949]: satip: epoll rem: fd=69 events=1
Oct 31 14:08:37 tvheadend[36949]: satip: SAT>IP DVB-C Tuner #1 (XXXX@UDP) - shutdown for /stream=1529/10521066
72
```

So fd 116 is not added (which is a problem I guess) and can not be removed because it was never added?

Do you need the epoll rem lines of a working tune on the same tuner? Whats happening now is that when zapping through the channels the tuner stops working at some point (I do not know when exactly that is) and when I stop zapping after about 50 channels its broken. Then I turn on the tracing and try to tune to a channel, tuner 1 fails and tuner 2 succeeds in this case. Where in this process should I look for the lines? I hope you dont need me to find the last working tune somewhere in this process?

I guess the only way this could happen is when tvhpoll.c:187 is true.....

#42 - 2019-10-31 21:29 - Jaroslav Kysela

Flole Systems wrote:

Shouldnt a new thread start for this anyways as I have "Fast Input Switch" disabled and all the old variables should be "forgotten"/reinitialized?

I looked at the stopping part of the tune shown above and there's this:
[...]

So fd 116 is not added (which is a problem I guess) and can not be removed because it was never added?

I don't think that 116 was not added. In that case tvhpoll_get_events() will return zero. The question is why we see "epoll del failed [No such file or directory]" . You may try to remove the break in tvhpoll_rem0(), but the question remains - what happened with fd 116? What's the state of this file descriptor when this error occurs? It's closed? ...

#43 - 2019-10-31 22:40 - Flole Systems

Are you sure about that? As you can see in my code above it is added to ev and ev is passed to tvhpoll_add but there is no line indicating that it was actually added to the epoll. That would also explain why there is data and the wait is not noticing that data and why removing it fails (cause its not added so it cant be removed) with said error:

```
ENOENT op was EPOLL_CTL_MOD or EPOLL_CTL_DEL, and fd is not registered with this epoll instance.
```

So I think the problem here is that adding it is already failing for some reason.

#44 - 2019-10-31 23:03 - Jaroslav Kysela

I thought that we are talking about this scenario:

```
stream #1:
  poll add ok
  poll rem error

stream #2:
  poll add not visible
  ....
```

If it's this:

```
stream:
  poll add not visible
  poll rem error
```

Then we should determine why the poll add trace is not visible (run) for the RTP file descriptor.

#45 - 2019-10-31 23:04 - Luis Alves

The only explanation for 116 not being added is that inside function "tvhpoll_add0", this condition is true:

```
if (oevents == events) continue;
```

<https://github.com/tvheadend/tvheadend/blob/master/src/tvhpoll.c#L187>

Not sure this "if" makes sense... what does it do?

#46 - 2019-10-31 23:15 - Luis Alves

Can you mod the code (ir use gdb) to print the contents of variables: events, oevents and fd ?

#47 - 2019-11-01 00:13 - Flole Systems

We are talking about this (done at a speed of about 2 steps per second, so fast zapping basically):

```
Stream #1 start
Stream #1 stop
Stream #1 start
Stream #1 stop
.....
Stream #1 start
No Data Coming in
```

For Debugging I do (same speed as above):

```
Stream #1 start
Stream #1 stop
Stream #1 start
Stream #1 stop
...
Stream #1 start
No Data Coming in
Stream #1 stop
Stream #1 start
No Data Coming in
Stream #1 stop
.....
Wait 10-20s
Stream #1 start
No Data Coming in
```

The if in that line is there to prevent double-adds if I see that correctly (If someone could confirm this that'd be great). I'm going to add a trace to it there so we can see if something is skipped

#48 - 2019-11-01 01:51 - Luis Alves

The code has 2 modes: kqueue and epoll.
I the default is epoll.
Have you tried to compile using kqueue?

```
./configure --disable-epoll --enable-kqueue
```

#49 - 2019-11-01 02:08 - Flole Systems

Kqueue is BSD/MacOS, epoll is Linux. The default should be selected based on what the system supports, Linux is kqueue incompatible and I won't set up a VM (or even switch the OS) for this test.

#50 - 2019-11-01 08:28 - Jaroslav Kysela

I also think that the poll layer have the fd registered (thus oevents == events condition is true).

Looking to the code again and it seems that the issue may be the 'goto done' in satip_frontend.c . The done label is bellow the tvhpoll_rem(), but it closes the rtcp/rtp UDP file descriptors. The 's' (start) pipe command may invoke this code path. Because epoll caches the file descriptor event subscription, it seems that the old events are in the cache.

Could you try latest ? (commit 25a50f75a07b656e380b4e9e2d61cbc6c7740e4b ?)

#51 - 2019-11-01 16:57 - Flole Systems

The original issue appears to be fixed now! Thank you so much for your help, its great to have support from someone who knows the code very well, without you it would be very hard to find bugs like this one! I really appreciate the time you spent troubleshooting this.

I will do some more checks later on and see if the other issue where no request is sent appears again but so far I didn't see that since the change you made.

#52 - 2019-11-01 17:43 - Luis Alves

It ended up having duplicate code.

Does the poll remove code really need to be before the "if (exit_flag)" ?
If not, we could simply have 1 poll remove block, after the "done:" label.

#53 - 2019-11-01 18:19 - Jaroslav Kysela

I know that the fix is not perfect. I need to review again the done label callers and try to optimize those bits. The `satip_frontend_shutdown()` should be called without RTP/RTCP file descriptors otherwise the `tvhpoll_wait()` loop will eat 100% CPU (there might be some data before the SAT>IP server replies to TEARDOWN).