

RTP Issues

About

RTP issues can be difficult to diagnose without a healthy amount of packet captures and other troubleshooting information.



This document was written in 2008 and is likely no longer applicable.

Devices with Issues

Sonus

Carrier Info

Carriers known to be using this device: L3, WilTel, Bandwidth.com, Broadvox, Global Crossing, XO, Qwest and T-Systems (a division of Deutsche Telekom).

- 19:40, 9 December 2008 (PST)
- Known to support exactly G.729 with rfc2833 OR G.711u with DTMF inband. If you're using FreeSWITCH then you can purchase a license for G.729 or use G.711u (PCMU) with DTMF inband. (also RFC2833 look below how)
- Dropped audio infrequently. Anthony discovered that they have a +500ms response time if you STOP streaming RTP data to the carrier.

Whatever hardware they use has a list of reasons to reset.

Causes:

- If your installation of FreeSWITCH wants to change timestamp base and send them mark bit, they reset with 2 seconds of silence.
- If you send RFC2833 on it's own timestamp base it resets.

Dropped Audio

If you are suffering from dropped audio, specifically regarding the first 2-3 seconds of a single audio stream then try the following:

conf/vars.xml

```
<!-- This will continue a steady stream of RTP to the Sonus device and your dropped audio will nearly vanish -->
<X-PRE-PROCESS cmd="set" data="send_silence_when_idle=400"/>
```

Take care when using this parameter, as it may suppress ringback tones by actually sending early media.

conf/sip_profiles/external.xml

```
<!--This will rewrite all new timestamps instead of passing them from the other leg.-->
<param name="rtp-rewrite-timestamps" value="true"/>
```

DTMF Problems

If you are having DTMF problems and **Sonus is in your media path**, you should make sure you are using the latest version of FreeSWITCH. As of FreeSWITCH revision 10744, FreeSWITCH auto-detects Sonus end-points and applies a hack to "fix" the issue. The hack is described as follows (from switch_types.h):

Sonus wrongly expects that, when sending a multi-packet 2833 DTMF event, the sender should increment the RTP timestamp in each packet when, in reality, the sender should send the same exact timestamp and increment the duration field in the 2833 payload. This allows a reconstruction of the duration if any of the packets are lost.

```
final_duration - initial_timestamp = total_samples
```

However, if the duration value exceeds the space allocated (16 bits), the sender should increment the timestamp one unit and reset the duration to 0.

Always sending a duration of 0 with a new timestamp should be tolerated but is rarely intentional and is mistakenly done by many devices. The issue is that the Sonus expects everyone to do it this way instead of tolerating either way. Sonus will actually ignore every packet with the same timestamp before concluding if it's DTMF.

Note that the version of FreeSWITCH which has this patch auto-detects Sonus, so you don't need to configure anything.

If you are going through a provider who uses Sonus **only as an SBC and not in your media path** FreeSWITCH may incorrectly identify your call as going through Sonus and actually corrupt your media stream's DTMF. You can override FreeSWITCH's auto-detection of end-user agents with a flag in profiles stored under `$FS_ROOT/sip_profiles/`. Just add:

Override FS user agent detection

```
<param name="auto-rtp-bugs" value="clear" />
```

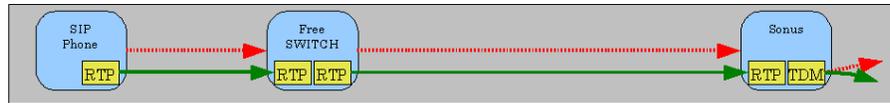
In some cases, you may have the reverse situation where Sonus or Cisco equipment sits in your media path, but FreeSWITCH can't auto-identify it. In that case, try these settings:

Cisco workaround

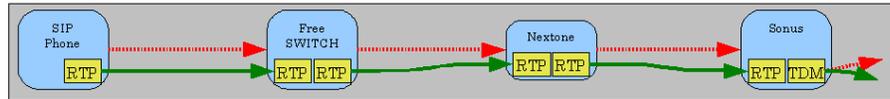
```
<param name="auto-rtp-bugs" value="CISCO_SKIP_MARK_BIT_2833" />
```

Sonus workaround

```
<param name="auto-rtp-bugs" value="SONUS_SEND_INVALID_TIMESTAMP_2833" />
```

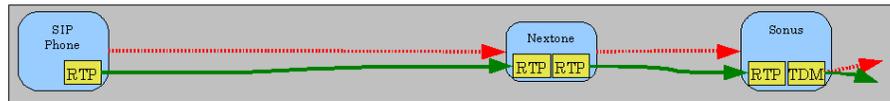


Call	GSX	Neg/Fix	LEG1	LEG2	Negotiated	DTMF/ IN	OUT	DTMF?
1	3	Neg.	PCMU, G723, PCMA, G729, t-e	PCMU, G729, t-e, CN	PCMU (t-e)*	OOB	INBAND	YES
2	3	Neg.	G729, G723, PCMU, PCMA, t-e	G729, G711U, t-e, CN	G729, t-e	OOB	OOB	YES
4	2	Neg.	PCMU, G723, PCMA, G729, t-e	PCMU, G729, t-e, CN	PCMU (t-e)*	OOB	INBAND	NO
3	2	Neg.	G729, G723, PCMU, PCMA, t-e	G729, G711U, t-e, CN	G729, t-e	OOB	OOB	YES
5	2	Neg.	PCMU, G723, PCMA, G729, t-e	PCMU, G729, t-e, CN	PCMU (t-e)*	OOB	INBAND	YES **



Call	GSX	Neg/Fix	LEG1	LEG2	Negotiated	DTMF/ IN	OUT	DTMF?
6	2	Neg.	PCMU, G723, PCMA, G729, t-e	PCMU, G729, t-e, CN	PCMU (t-e)*	OOB	INBAND	YES **

Just to make sure it wasn't just a FreeSWITCH to Sonus issue, I also performed the same tests with a Nextone and received the same result sets.



Call	GSX	Neg/Fix	LEG1	LEG2	Negotiated	DTMF/ IN	OUT	DTMF?
6	2	Neg.	PCMU, G723, PCMA, G729, t-e	PCMU, G729, t-e, CN	PCMU (t-e)*	OOB	INBAND	YES **

* Means the telephone-event is on the public response only.
 ** The base sonus packet service profile for GSX 2 was changed on the Fax Relay/Fallback section.

FreeSWITCH to Sonus testing paths

If your call to a Sonus still won't work

For those of you who have tried the auto-rtp-bugs and still cannot get DTMF to work for G711 negotiated calls, there appears to be an issue with the Sonus using Fax treatment which drops the out of band DTMF packets that come from FreeSWITCH (and other platforms). Some more in-depth information for Sonus users (at least for 5.0 and 5.1 users, I can't speak for those who may have the latest and greatest code revisions which might fix this) is below.

In the GSX Navigator (Sonus Insight), go to SonusProfilesPacket Service Profilesdefault. You may have some changes made to the Fax Treatment/Failure section. To resolve the DTMF you will have to change that to match the below examples.

An example of a base profile that will not work:

Priority	Codec	Pkt Size/ Code Rate	G.711 Law/ Send SID	DTMF Relay/ Remove	Modem Treatment/ Failure	Fax Treatment/ Failure
3	G711SS	20	LAWFROMCIR	NONE	G711FALLBACK	G711FALLBACK
		0BPS	DISABLED	DISABLED	CONTINUE	CONTINUE
4	G711	20	LAWFROMCIR	NONE	G711FALLBACK	G711FALLBACK
		0BPS	DISABLED	DISABLED	CONTINUE	CONTINUE

What you would need to change it to, to fix the DTMF:

Priority	Codec	Pkt Size/ Code Rate	G.711 Law/ Send SID	DTMF Relay/ Remove	Modem Treatment/ Failure	Fax Treatment/ Failure
3	G711SS	20	LAWFROMCIR	NONE	G711FALLBACK	FAXRELAYORG711FA
		0BPS	DISABLED	DISABLED	CONTINUE	CONTINUE
4	G711	20	LAWFROMCIR	NONE	G711FALLBACK	FAXRELAYORG711FA
		0BPS	DISABLED	DISABLED	CONTINUE	CONTINUE

The above profile will work if you are sending direct to the GSX using the dtg=TRUNKGROUP protocol. If you are routing your calls to the PSX, then you would make the appropriate changes to the Preferred Packet Service Profile that is attached to the trunk group that you are a) sending from or b) sending to.

Other Options

conf/sip_profiles/external.xml

```
<!-- This will use the async timer (this one you can also set to none if you don't want asynchronous RTP traffic) -->
<param name="rtp-timer-name" value="soft"/>
```

<!--

RTP_BUG_CISCO_SKIP_MARK_BIT_2833 = (1 << 0),

Some Cisco devices get mad when you send the mark bit on new 2833 because it makes them flush their jitterbuffer and the dtmf along with it.

This flag will disable the sending of the mark bit on the first DTMF packet.

RTP_BUG_SONUS_SEND_INVALID_TIMESTAMP_2833 = (1 << 1),

Sonus wrongly expects that, when sending a multi-packet 2833 DTMF event, The sender should increment the RTP timestamp in each packet when, in reality, the sender should send the same exact timestamp and increment the duration field in the 2833 payload. This allows a reconstruction of the duration if any of the packets are lost.

final_duration - initial_timestamp = total_samples

However, if the duration value exceeds the space allocated (16 bits), The sender should increment the timestamp one unit and reset the duration to 0.

Always sending a duration of 0 with a new timestamp should be tolerated but is rarely intentional and is mistakenly done by many devices.

The issue is that the Sonus expects everyone to do it this way instead of tolerating either way. Sonus will actually ignore every packet with the same timestamp before concluding if it's DTMF.

This flag will cause each packet to have a new timestamp.

RTP_BUG_IGNORE_MARK_BIT = (1 << 2),

A Huawei SBC has been discovered that sends the mark bit on every single RTP packet.

Since this causes the RTP stack to flush it's buffers, it horribly messes up the timing on the channel.

This flag will do nothing when an inbound packet contains the mark bit.

RTP_BUG_SEND_LINEAR_TIMESTAMPS = (1 << 3),

Our friends at Sonus get real mad when the timestamps are not in perfect sequence even during periods of silence.

With this flag, we will only increment the timestamp when write packets even if they are eons apart.

RTP_BUG_START_SEQ_AT_ZERO = (1 << 4),

Our friends at Sonus also get real mad if the sequence number does not start at 0.

Typically, we set this to a random starting value for your saftey.

This is a security risk you take upon yourself when you enable this flag.

RTP_BUG_NEVER_SEND_MARKER = (1 << 5),

Our friends at Sonus are on a roll, They also get easily dumbfounded by marker bits.

This flag will never send any. Sheesh....

RTP_BUG_IGNORE_DTMF_DURATION = (1 << 6),

Guess Who? ... Yep, Sonus (and who know's who else) likes to interweave DTMF with the audio stream making it take

2X as long as it should and sending an incorrect duration making the DTMF very delayed.

This flag will treat every dtmf as if it were 50ms and queue it on receipt of the leading packet rather than at the end.

RTP_BUG_ACCEPT_ANY_PACKETS = (1 << 7),

Oracle's Contact Center Anywhere (CCA) likes to use a single RTP socket to send all its outbound audio.

This messes up our ability to auto adjust to NATTED RTP and causes us to ignore its audio packets.

This flag will allow compatibility with this dying product.

RTP_BUG_GEN_ONE_GEN_ALL = (1 << 8),

Some RTP endpoints (and by some we mean *cough* _SONUS_!) do not like it when the timestamps jump forward or backwards in time.

So say you are generating a file that says "please wait for me to complete your call, or generating

```
ringback"
```

```
    Now you place and outbound call and you are bridging. Well, while you were playing the file, you were generating your own RTP timestamps.
```

```
    But, now that you have a remote RTP stream, you'd rather send those timestamps as-is in case they will be fed to a remote jitter buffer.....
```

```
    Ok, so this causes the audio to completely fade out despite the fact that we send the mark bit which should give them heads up its happening.
```

```
    Sigh, This flag will tell FreeSWITCH that if it ever generates even one RTP packet itself, to continue to generate all of them and ignore the actual timestamps in the frames.
```

```
RTP_BUG_CHANGE_SSRC_ON_MARKER = (1 << 9),
```

```
    By default FS will change the SSRC when the marker is set and it detects a timestamp reset.
```

```
    If this setting is enabled it will NOT do this (old behaviour).
```

```
RTP_BUG_FLUSH_JB_ON_DTMF = (1 << 10)
```

```
    Flush jitterbuffer when getting RFC2833 to reduce bleed through
```

```
Examples usage in profile parameters:
```

```
<param name="manual-rtp-bugs" value="
```

```
SEND_LINEAR_TIMESTAMPS|START_SEQ_AT_ZERO|NEVER_SEND_MARKER|IGNORE_DTMF_DURATION|ACCEPT_ANY_PACKETS|GEN_ONE_GEN_ALL"/>
```

```
-->
```

Voice Operator Panel (VOP)

VOP-Softclients don't like changes in the RTP timestamp, e.g. when a REFER to the VOP-client occurs and the RTP-stream is replaced by the stream from a different call with a different timestamp. This leads to one-way audio for the callee, the VOP Softphone doesn't process the incoming RTP packets with the "new" timestamp and the callee will not hear the caller.

Setting [rtp-rewrite-timestamps](#) to True in the affected SIP-profile eventually solves the problem.

See also

For more information on these and other Sonus issues:

- http://www.submityoursip.com/wiki/Sonus_NBS or http://web.archive.org/web/20100121175135/http://www.submityoursip.com/wiki/Sonus_NBS
- <http://blog.krisk.org/2009/02/update-youve-been-waiting-for.html>
- <http://www.freeswitch.org/node/160>