Proxy Media

About

Proxy Media mode puts FreeSWITCH in a "transparent proxy mode" for the RTP streams. The RTP streams still pass through FreeSWITCH (unlike bypass media mode) by using a static all-purpose codec that cannot be decoded. As a consequence any attempts to play a file or record the call will result in an immediate hangup of the call. Its main use was initially to make T.38-passthrough possible because by ignoring the codecs FreeSWITCH could switch to T.38 mode without actual support for T38. FreeSWITCH now supports T.38 and T.38 passthrough properly so this feature is less useful. The other consequence of this feature is that all reads on the RTP stream are blocking and there is no async RTP possible so audio problems can arise if you are not connected to a reliable RTP endpoint on the other side of the stream.

Common misconceptions (READ THIS)

Many people seem to think that proxy_media is the opposite of bypass_media and is therefore required to proxy media through FS. This is not true. FreeSWITCH has 3 media handling modes:

- **Default**: media flows through FS, full processing options
  - RTP proxied by FreeSWITCH
  - FreeSWITCH controls codec negotiation
  - If endpoints agree on same codec, no transcoding is performed
  - All features enabled - recording, DTMF interception, etc, etc

- **Proxy**: media flows through FS, no media processing options
  - RTP proxied by FreeSWITCH {c= modified, that's it}
  - FreeSWITCH has no control or even understanding of other SDP parameters (pass through)
  - Endpoints *MUST* agree on same codec because FreeSWITCH can't help them
  - Virtually no features available

- **Bypass**: media flows around FS directly between endpoints, no media processing options
  - RTP *NOT* proxied by FreeSWITCH
  - FreeSWITCH has no control over anything SDP related - it's completely pass through from one leg to next (including c=)
  - Endpoints must agree on same codec because FreeSWITCH doesn't even see the media or SDP
  - Virtually no features available

Usually if you want to send media through FS so the endpoints don’t talk to each other directly (topology hiding) you want the Default mode which accomplishes this but still gives you all the media processing abilities.

Why you almost certainly don’t want to use it

- Anything that requires reading or modifying media will NOT work in Proxy Media mode. This includes:
  - Transcoding
  - Handling of inband DTMF (converting in either direction)
  - Conference, record, intercept, IVR etc
  - Pretty much any feature you can think of that involves media
  - Bridging between endpoints that don’t use RTP for media (e.g. TDM, Skype)
  - It is no longer required for T38, as that is supported by mod_spandsp.

Pretty much the only feature that will still work is bridge.

Why you might want to use it

- Allows FreeSWITCH to handle codecs that it does not officially support (e.g. G.729 without licensed codecs for FreeSWITCH)
- You wish to develop an OSTD compliant soft switch that allows the media stream to flow through a proxy unaltered, which ensures ZRTP handshake data will persist.
How to enable it

Set proxy_media=true before the bridge.

```xml
<action application="set" data="proxy_media=true"/>
```

Alternatively, configure the SIP profile to use proxy media by default:

```xml
<param name="inbound-proxy-media" value="true"/>
```

(see example in conf/sip_profiles/internal.xml)

Important note

You can't "proxy_media" two outbound legs.

This means that if you originate a call through some API ('originate', for example), this will NOT work.

In fact, the 1st leg will answer before the 2nd leg is called.

The 1st one will share the SDP with FreeSWITCH but will not be able to re-share with the 2nd one.

FAQs

I get "Codec PROXY PASS-THROUGH encoder error" message

Enable late negotiation.

If you're trying to do anything that processes media, don't use proxy_media.

Proxy media doesn't work

Enable late negotiation.

Proxy media does not work with ZRTP trusted MITM

ZRTP call between 2 csipsimple UAs with Trusted mitm works. But when proxy-media is enabled the UAs ignore the zrtp-hash. In that case you should enable inbound-zrtp-passtrhu

See Also

- Codecs
- Codec Negotiation