

# [webrtc] RTX的处理 - 一 水

以前笔记，整理

webrtc中默认开启rtx用于丢包重传，rtx的介绍可以参考rfc4588，  
<https://tools.ietf.org/html/rfc4588#section-4>

rtx使用额外的ssrc传输，ssrc在sdp中会标识出来。

```
←a=rtpmap:97 rtx/90000  
←a=ssrc-group:FID 2736695910 239189782
```

类似这样。

一个RTX包，在turnserver中是这样的，原始udp数据->turn/stun协议头->RTP header1 ->RTP header2

在RTP header1中根据payload type进行区别RTP、RTX数据，如果是RTX的话，需要srtp解出后面的数据，再解析。

在客户端中，RTX封包的关键函数是：

```
https://code.google.com/p/webrtc/source/detail?r=4692  
Channel::IsPacketRetransmitted  
Channel::HandleRtxPacket  
rtp_payload_registry->IsRtx  
RTPPayloadRegistry::RestoreOriginalPacket 移除RTX头, 还原原始的RTP
```

链接是，webrtc加入rtx的issue

目前webrtc的服务器，例如licode，janus是禁止rtx的。以避免复杂的处理。

以前笔记，整理

webrtc默认使用UDP传输，但是也可以通过TCP传输。

使用tcp传输，需要服务器中转，turnserver，licode，janus之类的服务器。

1. 如果使用turnserver的话，只需要客户端保留 relay&&tcp类型的candidate，其他都抛弃即可
2. 如果使用的是licode，janus之类的服务器的话，默认是不支持tcp的。

因为它们底层使用都是libnice这个开源的ICE库，libnice在比较新的版本中才支持tcp的

所以需要自己编译新版本的libnice，然后重编译服务器

参考libnice的文档 <https://nice.freedesktop.org/libnice/NiceAgent.html#NiceAgent--ice-tcp>

### 3. libnice新版也支持tcp 心跳检测

libnice 0.1.8以后支持keepalive-conncheck

```
gboolean keepalive = TRUE;
```

```
g_object_set(G_OBJECT(agent), "keepalive-conncheck", keepalive, NULL);
```

就是超时时间有点长。

最大超时时间在 time.h中

```
total_timeout = initial_timeout * (2^(max_retransmissions + 1) - 1);
```

20s-80s左右超时NICE\_COMPONENT\_STATE\_FAILED