

WebRTC中丢包重传NACK实现分析 - 简书

简书 jianshu.com/p/a7f6ec0c9273

在WebRTC中，前向纠错(FEC)和丢包重传(NACK)是抵抗网络错误的重要手段。FEC在发送端将数据包添加冗余纠错码，纠错码连同数据包一起发送到接收端；接收端根据纠错码对数据进行检查和纠正。RFC5109[1]定义FEC数据包的格式。NACK则在接收端检测到数据丢包后，发送NACK报文到发送端；发送端根据NACK报文中的序列号，在发送缓冲区找到对应的数据包，重新发送到接收端。NACK需要发送端发送缓冲区的支持，RFC5104[2]定义NACK数据包的格式。

本文在研究WebRTC源代码的基础上，以Video数据包的发送和接收为例，深入分析NACK丢包重传机制的实现。主要内容包括：SDP协商NACK，接收端丢包判定，NACK报文构造、发送、接收和解析，RTP数据包重传。下面分别详细论述之。

1 SDP协商NACK

NACK作为RTP层反馈参数，和Video Codec联系在一起。WebRTC在初始化阶段，创建PeerConnectionFactory对象，在该对象中创建MediaEngine，其中的VideoEngine为WebRtcVideoEngine2。该对象在构造时，会收集本端支持的所有Video Codec，NACK作为Codec的属性被一起收集。在接下来的SDP协商过程中，NACK属性被协商到Offer/Answer中，如图1所示。

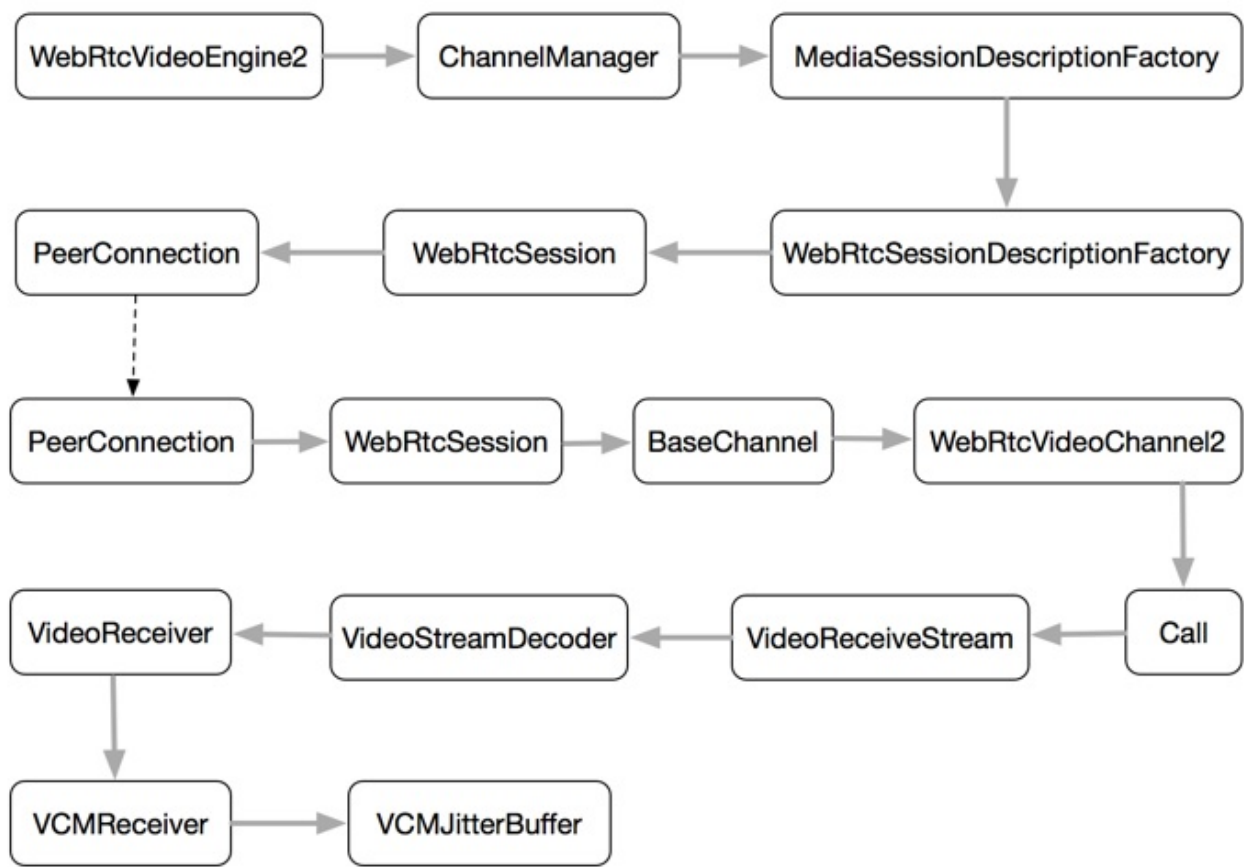


图1 SDP协商NACK及作用于Video JitterBuffer

PeerConnection在CreateOffer时，收集本端的会话控制信息、音视频Codec信息和网络信息等内容。视频Codec信息从WebRtcVideoEngine2中获取。最后本端Offer形成SDP报文，经过PeerConnection对象发送到网络。

接收端在收到Offer之后，首先调用SetRemoteDescription，根据本地配置信息向下创建VideoReceiveStream对象，本地NACK配置信息会最终到达VCMJitterBuffer。接着PeerConnection调用CreateAnswer，生成Answer；根据Offer中的Codec信息和本端支持的Codec信息，最终选定双方都支持的Codec集合。最后用生成的Answer作为

参数调用SetLocalDescription，根据Answer中的Video Codec信息向下重新创建VideoReceiveStream对象。NACK信息向下传递最终到达VCMJitterBuffer，在这里设置NACK相关参数。这些参数在接收RTP数据包过程中发挥作用，比如判断丢包、是否发送NACK报文等。Answer发回发送端时，发送端调用SetRemoteDescription执行同样的设置流程。

2 接收端丢包判定

Video接收端丢包判定在Worker线程中进行。RTP数据包到达接收端后，经过RTP模块到达VCM模块的JitterBuffer对象，最终调用VCMJitterBuffer的InsertPacket函数对数据包进行缓存和重排。

VCMJitterBuffer把丢失RTP数据包的序列号存储在集合missing_seq_nums中。对于本次从RTP模块到来的数据包，标记其序列号为seq1，而上次到达数据包的序列号为seq2。如果seq1 > seq2，则表示seq1顺序到达，标记(seqnum2, seqnum1)区间内的数据包为丢失状态，将其存储到missing_seq_nums集合中。注意这里的丢失状态是暂时的，如果下个数据包到达时有seq1 < seq2，则表示数据包乱序到达，则把missing_seq_nums中小于seq1的序列号都删除掉。

在更新missing_seq_nums集合时，如果集合中存储的序列号超过预设的容量，则通过调用RecycleFramesUntilKeyFrame()不断丢包来减少集合中的序列号，直到集合中的序列号总数低于预设容量值。

3 NACK报文发送和接收

接收端的NACK报文构造和发送工作在ModuleProcessThread线程中周期性完成。过程如图2所示。

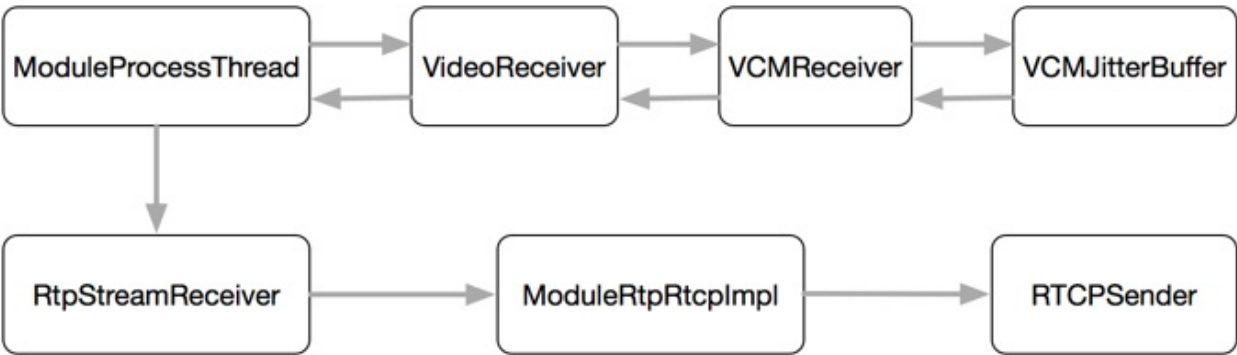


图2 NACK报文构造和发送

ModuleProcessThread线程周期性调用VideoReceiver::process函数，该函数通过VCMReceiver调用VCMJitterBuffer::GetNackList，从missing_seq_nums集合中得到过去一段时间内丢失RTP数据包的序列号。然后调用RtpStreamReceiver::ResendPackets函数。调用流程最终会到达RTCPSender::SendRTCP，发送类型为NACK的RTCP报文。

NACK报文是类型为205的RTCP 扩展反馈报文，在RFC4585中定义[4]。具体格式如图3所示：

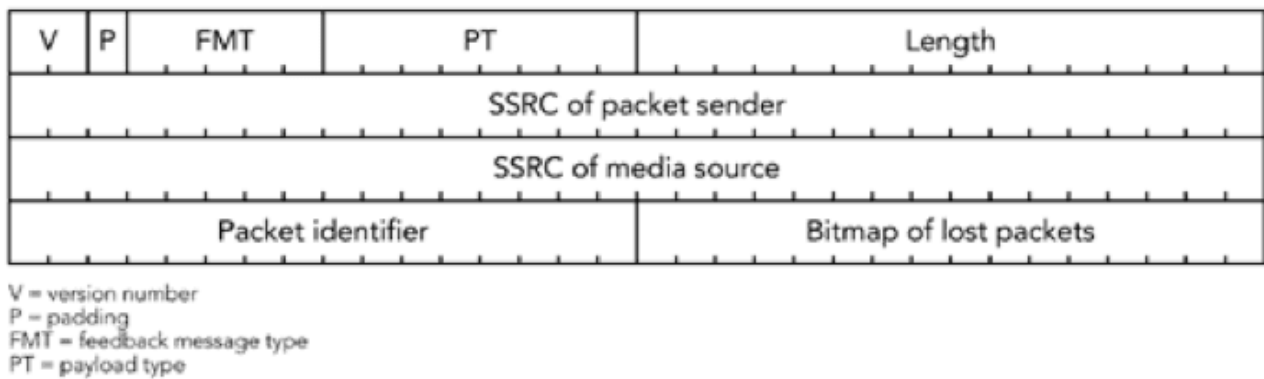


图3 NACK报文格式

其中PT = 205，FMT = 1，Packet identifier(PID)即为丢失RTP数据包的序列号，Bitmao of Lost Packets(BLP)指示从PID开始接下来16个RTP数据包的丢失情况。一个NACK报文可以携带多个RTP序列号，NACK接收端对这些序列号逐个处理。

NACK报文构造完成以后，发送到网络层。NACK报文是RTCP报文的一种，因此其发送、接收和分析遵循RTCP报文处理的一般流程。这部分内容可参考文档[3]。

4 RTP数据包重传

接收端在接收和解析NACK报文后，通过回调机制处理各种类型的RTCP报文，对于NACK报文，会调用RTPSender重新发送RTP数据包，如图4所示：

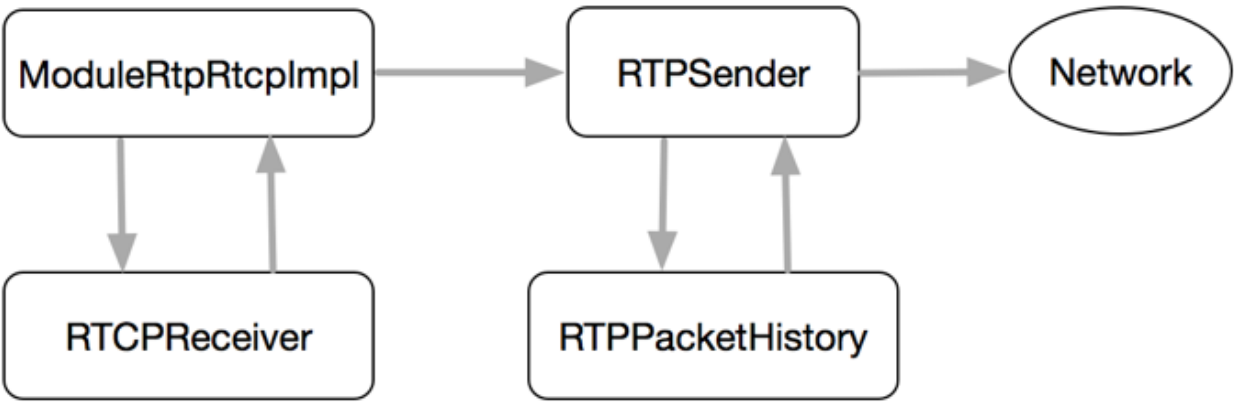


图4 发送端数据包重传

RTCPReceiver在解析RTCP之后，得到RTCP报文的描述结构，然后通过回调进行报文语义处理。NACK报文会被发送到RTPSender进行处理。RTPSender根据NACK报文中包含的序列号，到RTPPacketHistory缓存中查找对应的RTP数据包。如果找到，则把数据包发送到网络。

至此，一个完整的NACK报文回路完成，丢失的RTP数据包会重新发送到接收端。

5 总结

本文深入分析了WebRTC内部关于丢包重传(NACK)的实现细节，对NACK的SDP协商、丢包判定和重传进行深入研究，为继续学习掌握WebRTC的QoS机制奠定基础。

参考文献

- [1] RFC5109 - RTP Payload Format for Generic Forward Error Correction.
- [2] RFC5104 - RFC 5104 - Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)
- [3] WebRTC中RTP/RTCP协议实现分析 - <http://www.jianshu.com/p/c84be6f3ddf3>
- [4] RFC4585 - Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)