# **一、**[信道](https://so.csdn.net/so/search?q=%E4%BF%A1%E9%81%93&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/qw225967/article/details/_blank)**保护措施背景介绍**



  我们都知道，随着半导体集成技术的发展，智能移动设备已经全面普及，进入了移动互联网时代。在移动互联网时代中，大量的网络传输都发生在无线网络环境中，在方便用户使用的同时也带来一些问题。例如：带宽抖动、传输延迟、数据丢失与错误、异构性。这几个因素极大的影响了用户的使用体验。与其他数据传输量较少的应用相比，网络直播类应用对这三个因素的要求更高。如果说带宽与传输限制还能通过低[码率](https://so.csdn.net/so/search?q=%E7%A0%81%E7%8E%87&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/qw225967/article/details/_blank)来勉强满足用户的观看需求，那极易丢包以及异构化的环境对网络直播的影响是致命性的。  
因此，为了保证信道传输的安全、高质量，发展出了许多保护措施。比较具有代表性的三大措施为：丢包重传、码率自适应、前向纠错。

## **1.1、ARQ丢包重传**

  ARQ 全称 Automatic Repeat reQuest，中文意译为丢包重传，是一种通过重传关键数据包来纠错的信道保护算法。  
具体地来说，发送端给每一个数据包都植入顺序号码和时间戳，顺序号码代表被发送数据包的顺序，允许接收端可以通过监测顺序号码来发现丢包事件；时间戳代表语音视频数据包[解码](https://so.csdn.net/so/search?q=%E8%A7%A3%E7%A0%81&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/qw225967/article/details/_blank)的时间点。发送端发送数据包后，如果接收端没有收到，接收端将会通过 信令 信道发送一个重传请求。发送端维护一个缓冲队列，当收到重传请求的时候将会重传数据包。接收端也会维护一个缓冲队列，等待尚未收到的数据包以及对已经收到的数据包进行排序。在解码的 deadline 到来之前，接收端把缓冲区的数据包交给解码器进行解码。在解码 deadline 的时间点，接收端要么已经收齐了预期的数据包，要么已经决定放弃继续等待。

## **1.2、ABC码率自适应**

  ABC 全称 Adaptive Bit-rate Control，中文意译为码率自适应，是服务端和推流端协作控制码率来自动适应网络环境变化的技术。码率自适应的目的是为了对抗弱网环境。在网络好的情况下，适当提高码率，提高语音视频的质量和降低延迟；在网络差的情况下，适当降低码率，保障语音视频通话的可用性和流畅性，适当牺牲音画质量。

## **1.3、FEC前向纠错**

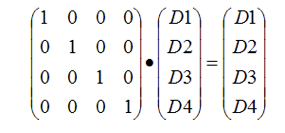
  FEC 全称是 Forward Error Correction，中文翻译为前向纠错，是一种通过增加冗余数据对丢失的数据包进行恢复的信道编码算法。具体地说，由发送端对原始数据进行 FEC 编码，生成冗余奇偶校验数据包，原始数据和冗余数据包合并称作 FEC 数据块，原始数据包和冗余数据包的数量比例是固定的。发送端发送 FEC 数据块。接收端接收到 FEC 数据块后，通过冗余数据包和原始数据包来恢复出丢失或者出错的数据包。  
  FEC 编解码算法目前比较成熟的为： RS(Reeds-Solomon) 算法、Raptor 算法和 Tornado 算法。  
  但罗马非一日建成，我们下面将从最基本的前向纠错算法来进行介绍。

# **二、FEC基本原理介绍**

[FEC算法\_cloudfly\_cn的博客-CSDN博客\_fec算法](https://blog.csdn.net/u010178611/article/details/82656838)

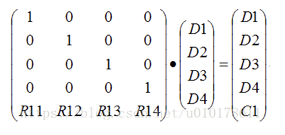
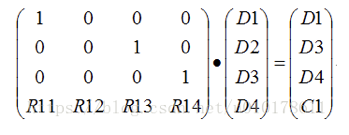
  静态的FEC指的是冗余度、保护程度都不随着网络变化而变化的前向纠错算法，而动态的FEC指的是保护算法会随着网络的波动而不断调整保护策略的前向纠错算法，下面将从静态FEC开始介绍，逐步深入。

## **2.1、静态FEC基本原理**

  首先，假设我们有数据包D1、D2、D3、D4，以最简单的对角01矩阵为掩码表可得：  


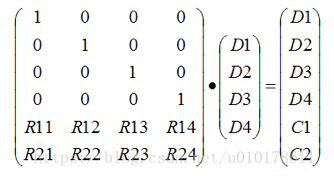
  可知，换算的结果还是他们本身。

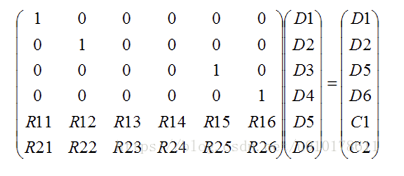
### **2.1.1、一阶冗余**

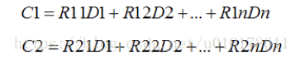
  那么什么是一阶冗余呢？其实很简单，所谓一阶冗余算法，就是每n个数据插入一个冗余数据（也即FEC编码组长度为n+1）；这n个数据和其对应的冗余数据构成一组数据，这组数据中丢掉任何一个数据都可以通过另外n个数据恢复出来。如下图，我们加入了一行冗余数据R11、R12、R13、R14来进行换算。  
  
  结果多出了一个C1冗余数据包。  
  当我们数据D2在传输中丢失时将出现以下情况：  


  这时候机智的我们将可以用C1进行计算反推出D2。具体的推算过程如下：  
IMG_260  
IMG_261

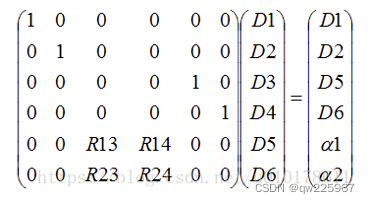
### **2.1.2、二阶冗余**

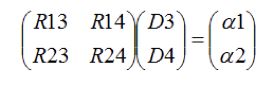
  相信有了以上的基础，不难看出二阶冗余就是再多加入一行冗余数据来进行计算的。  


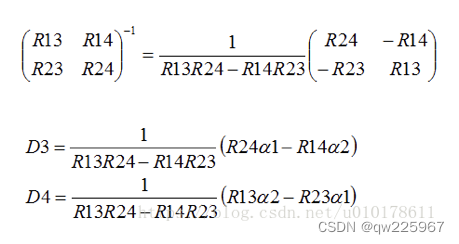
  当在传输中丢失了D3、D4时也就有了以下的矩阵等式。  


  可推算出C1、C2为：  


  由于D3、D4丢失，因此我们对矩阵进行消元，并令剩下的元素进行矩阵运算后的结果为 α1、α2。  
IMG_265

  那么我们的矩阵就变成了这样：  


  事实上就是解以下方程：  


  最终得到了D3、D4为以下结果：  


  其实根据以上的方法，我们还可以推出更高阶冗余的算法，在此就不赘述了。

## **2.2、动态FEC介绍**

虽然静态FEC前向纠错可以有效的提高数据可靠性，但由于其冗余模式固定，无论网络状态如何，都会消耗一定的网络带宽，导致在正常网络状态下减少了音视频传输的有效带宽。因此，自适应向前纠错应运而生。

### **2.2.1、动态FEC的发展**

  自适应前向纠错可以作这样表述：假设R(t)代表传输时刻 t 的网络带宽，而Rs(t)和Rfec(t)分别表示分给音视频数据和FEC冗余包的带宽，则需要在两者之间选择一个最优点。  
  早在2000年的ICC国际会议上，就有研究人员提出了多播自适应FEC算法（MA-FEC），该算法在确保维持Qos的条件下根据分组丢失率来动态决定冗余包数量，提供了可权衡传输开销和Qos保障的最佳权衡。随后，学者们又提出了各式各样的FEC前向纠错算法，例如根据四种不同网络制定的自适应网络算法。文献给出了四个自适应FEC协议：XRO-FEC、RS-FEC、UI-FEC、Pro-FEC。在不同网络分别使用这四种协议进行计算，实现动态切换。虽然国内外学者提出了大量的动态FEC示例，但完整的工程运用较少。直到google公司开发出[WebRTC](https://so.csdn.net/so/search?q=WebRTC&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/qw225967/article/details/_blank)，并在其中运用了自研的FEC技术，打开了FEC应用的大门。

### **2.2.2、WebRTC中的FEC**

  Webrtc中的fec分为两部分UlpFEC和FlexFEC；  
  UlpFEC为Uneven Level Protection FEC的缩写，非均衡级别保护向前纠错。它可以针对不同数据包提供不同级别的保护，从而对重要数据包进行更多保护。  
  FlexFEC为灵活性向前纠错，使用的是交织编码，交织编码复杂性高，但保护性更好，而且能有效解决（RFC 2733、RFC 5109和SMPTE 2022-1）中遇到的伸缩性问题。

# **三、业务相关的FEC（上行弱网时推流使用FEC）**

  在直播业务中，主播推流的质量有着非常重要的影响。我们应保证主播在上行弱网环境下，即使出现了丢包也能有较好的推流质量。使用WebRTC中的UlpFEC可以针对数据的重要性进行保护，有效的降低丢包对整个播放链路的影响。  
UlpFEC可以采用两种报文发送方式：1）使用独立RTP发送；2）封装在RED报文中随媒体数据发送。（WebRTC采用的是第二种方式）

****应用流程保密省略…****

  总之效果比较明显：  


# **四、结语**

  弱网优化是一个长期的、系统性的问题，单一的信道保护措施并不能达到最好的效果，就类似于自研弱网优化中使用的NACK+FEC的方案。我们通过多种信道保护措施相结合，达到了整体传输质量的提升。但系统中任何一个部分都尤为重要，因为任何一块短板都会对性能造成极大的影响，因此每一个细节、每一个技术我们都应该做到了如指掌。对FEC的分析使我们了解了基本的前向纠错原理，该类业务的拓展离不开基础的构筑。像FEC这样的技术之砖我们会一块一块堆砌，直到成为一面坚固的墙来阻挡任何弱网环境的侵蚀。

参考文献与网络博客：  
•[1]Priority encoding transmission. Proceedings of the 35th Annual Symposium on Foundations of Computer Science  
•[2]Modeling and Optimization of PET-Redundancy Assignment for MPEG Sequences  
•[3]Hybrid error control mechanism for video transmission in the wireless IP networks  
•[4]An MPEG performance model and its application to adaptive forward error correction  
•[5]A model for MPEG with forward error correction and TCP-friendly bandwidth.  
•[6]Adaptive FEC-Based Error Control for Internet Telephony  
•[7]QoS-sensitive transport of real-time MPEG video using adaptive forward error correction

https://blog.csdn.net/u010178611/article/details/82656838?utm\_medium=distribute.pc\_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-8.nonecase&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-8.nonecase  
https://tools.ietf.org/html/rfc5109  
https://tools.ietf.org/html/rfc2733  
https://ieeexplore.ieee.org/document/7291470  
https://tools.ietf.org/html/draft-ietf-payload-flexible-fec-scheme-03​