



实时音视频抗丢包的实践

Matthew (何永德)



好视通 音视频开发工程师

工作以来，一直从事通信相关行业

擅长音视频服务器性能的设计和优化

擅长实时音视频系统qos的设计和优化



内容介绍

分享“好视通”通过fec、nack、带宽自适应等技术，用来对抗网络丢包，达到在复杂的网络环境中保证音视频流畅性而做的一些工程化实践。

无线时代，丢包成为常态



有线物理层丢包率小于0.0001%



公共环境下wifi/4G可达到4%以上



移动流量占比在2021年预计达80%

抗丢包三板斧

--前向纠错FEC

--丢包重传ARQ

--拥塞控制CC



负责进攻（丢包恢复）

负责防守（控制码率）



FEC

XOR

- Data a,b,c,d
- Let $e = \text{XOR}(a,b,c,d)$
- Then
 - $a = \text{XOR}(b,c,d,e)$
 - $b = \text{XOR}(a,c,d,e)$
 - $c = \text{XOR}(a,b,d,e)$
 - $d = \text{XOR}(a,b,c,e)$

一般使用FEC生成矩阵。



$$r \leq k$$

当“丢包” $\leq r$, 可恢复

时延受分组长度影响，带宽利用率低，做不到100%恢复。

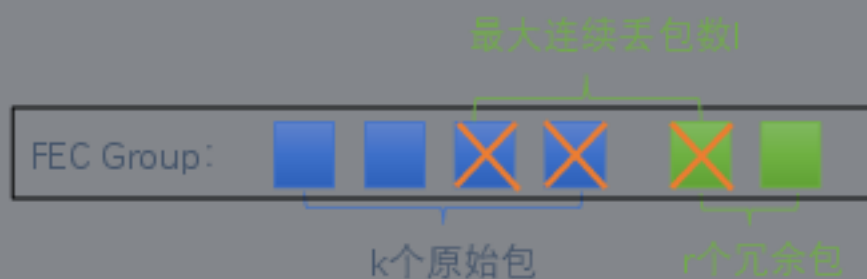
R的选择



我们将K个原始包和R个冗余包称为一个FEC分组，分组内只要丢失的包数目小于R个，就能够恢复。

R影响恢复能力，R根据网络丢包率来计算。需要满足 $R/(R+K) > \text{网络丢包率}$

FEC分组K的选择



由于 $R \leq K$, 而 $L \leq R$, 所以 $K \geq L$ 即K要大于最大连续丢包数

K也不能过大, 否则时延会高

小码流优化-缩小包长



码率较小时，适当缩小fec包长，也就缩小了包间隔时长，缩短丢包判断时间



缩短了分组的传输耗时，缩短丢包恢复时间

小码流优化-控制分组长度

--分组过长导致在丢包时加剧时延

小码率时，包间隔都挺长



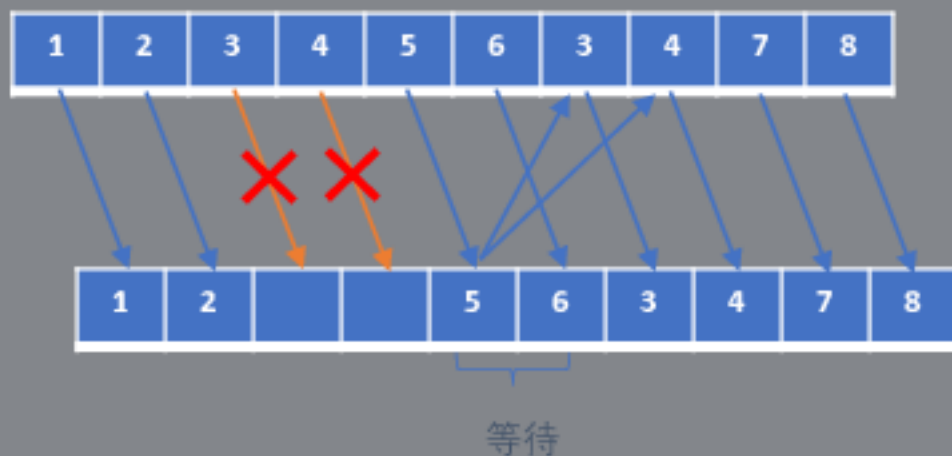
引入的延时
增加了一倍



但是太短的FEC分组，对抗连续丢包能力会明显下降

ARQ

ARQ (Automatic Repeat-request) 自动重传请求

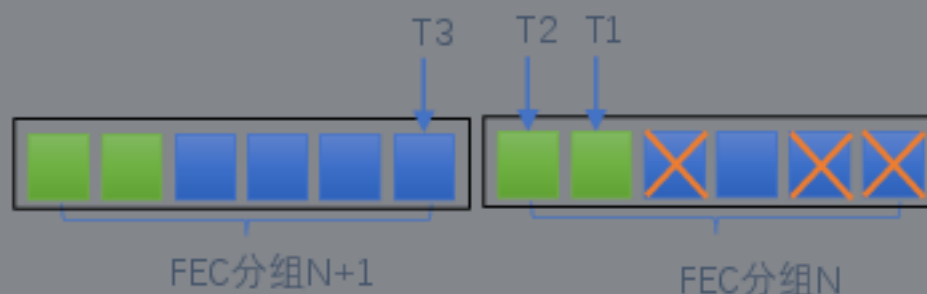


受rtt影响，带宽利用率高，能100%恢复

优化丢包判断的精度

-- 首次丢包请求时间

- T1: 包序号不连续
- T2: 序号差达到一定跨度
- T3: 收到下一个分组包后



如何更准确的判断一个包是乱序了还是丢包了？

-- 重复请求时间

ARQ和FEC的协作—上下层协议

FEC

ARQ架空FEC，无差别重传。

--浪费带宽

--浪费时间

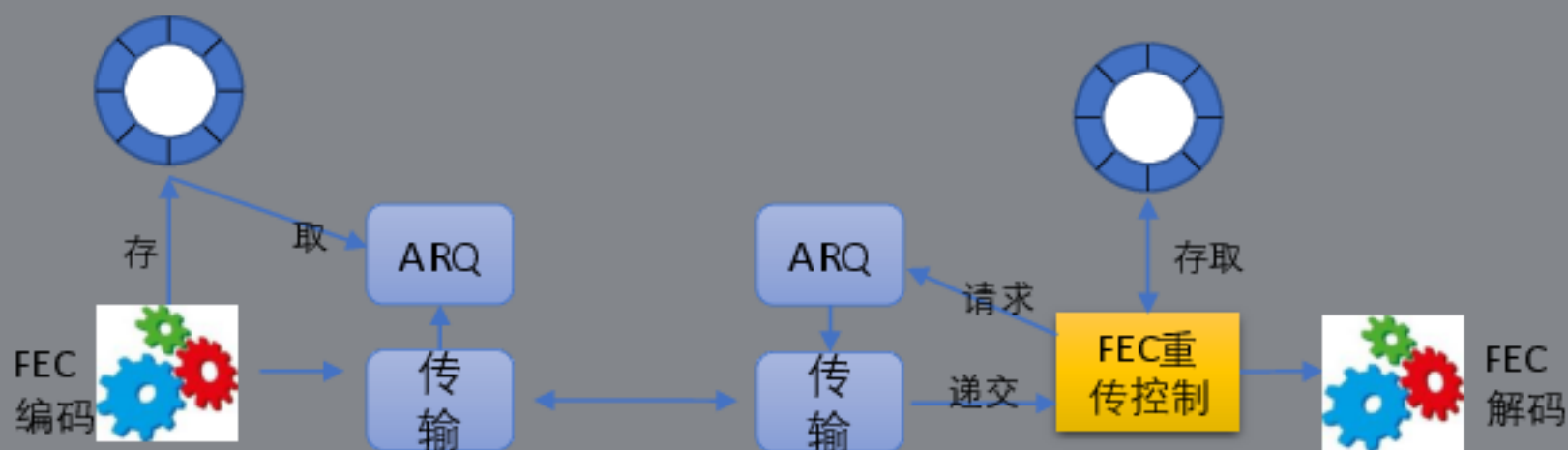
ARQ

ARQ

一旦FEC恢复不了，冗余包全部作废，

FEC

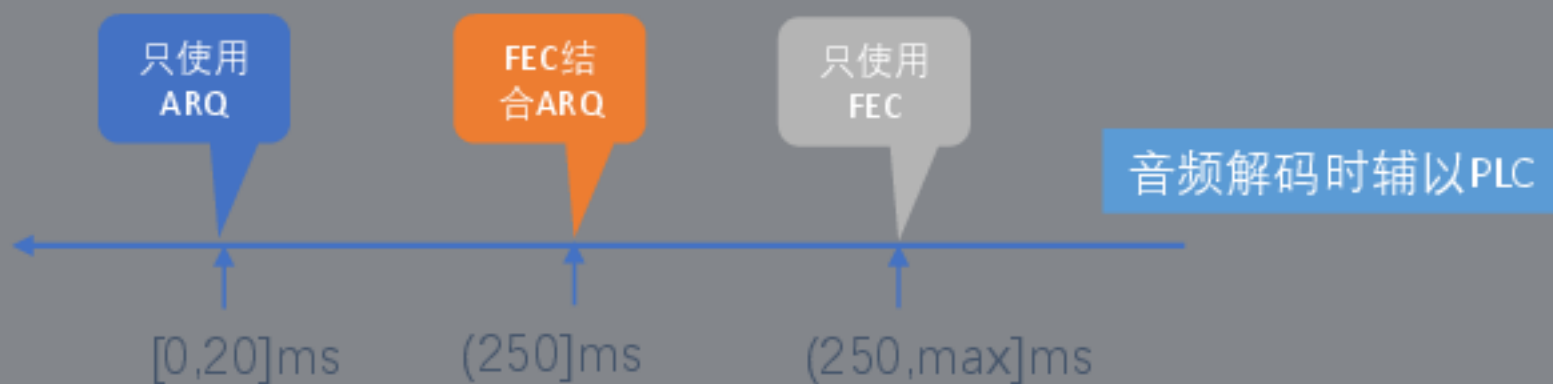
ARQ和FEC的协作—融合一体



ARQ和FEC融合为一个协议，共用协议头，ARQ使用FEC的分组号和组内标号来定位要重传的包。

FEC、ARQ的选择

RTT时间和使用策略



小码率时，把只使用ARQ的范围适当变大些。

拥塞控制--带宽估计

“带宽估计”不是估计带宽！！

它的核心诉求是：

当网络已经发生拥塞，在最短时间发现，并持续评估拥塞情况，
采取措施，降低拥塞造成的影响

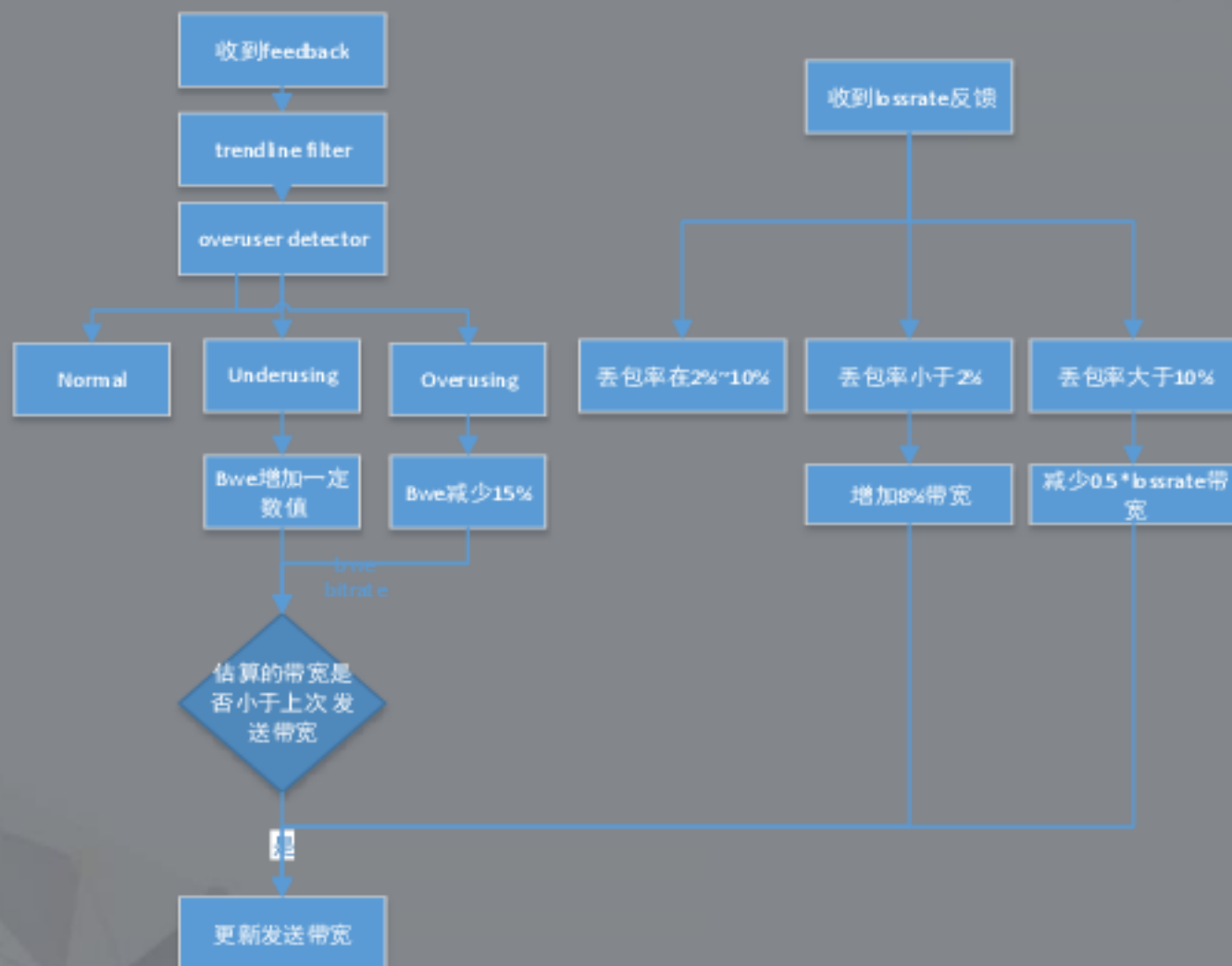
基于两种事实，产生了两种方法：

拥塞会导致较严重的丢包

统计丢包情况来评估拥塞

拥塞会导致包间隔拉长

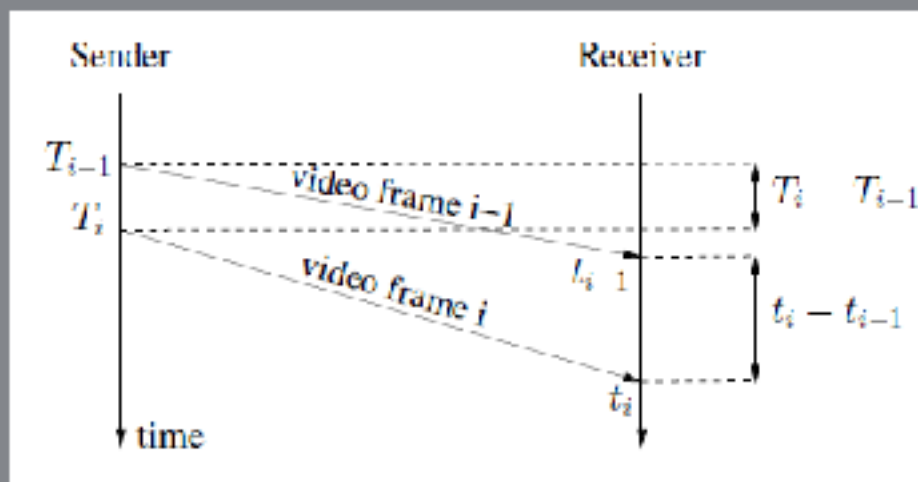
观察包间隔变化趋势来评估拥塞



延迟梯度

通过观察梯度值，可以知道网络是否拥塞。

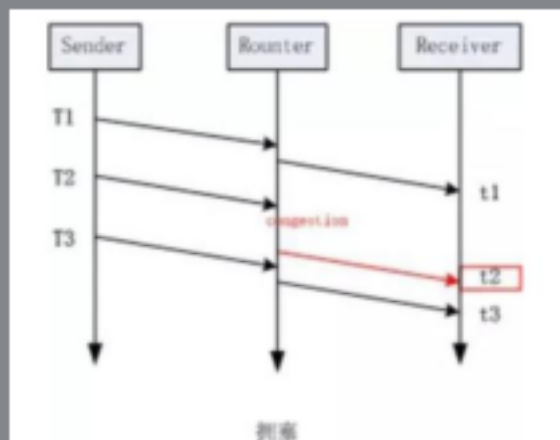
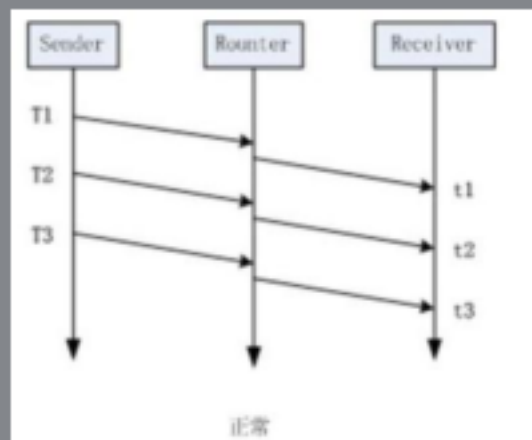
通过观察梯度值的变化趋势，可以感知到拥塞的变化情况。



$$d_m(t_i) = (t_i - t_{i-1}) - (T_i - T_{i-1})$$

延迟梯度，即包间隔的差值

延迟梯度判断拥塞

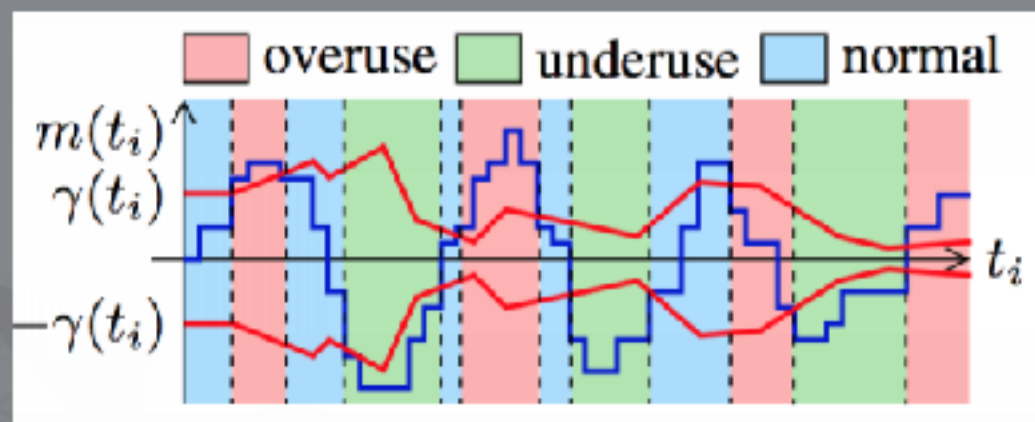
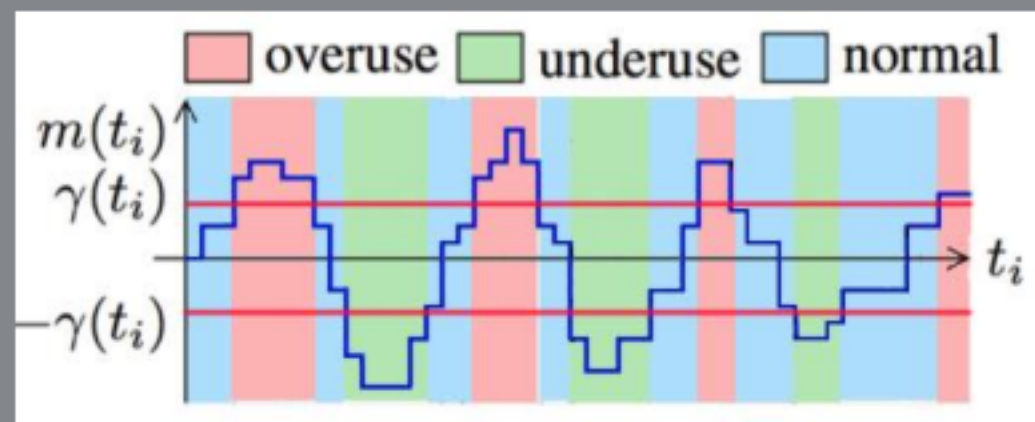


左边图正常状态，延迟梯度为0

右边图发生拥塞，包2计算的延迟梯度会变大

梯度值要去抖，常见的是采用 Kalman，以及GCC最新 Trendline

获得网络负载情况



```

1 if  $|m_i| > \gamma$  then
2   if  $m_i > 0$  then
3      $t_{OU} \leftarrow t_{OU} + \Delta T$ ;
4     if  $t_{OU} > \bar{t}_{OU}$  then
5       if  $m_i \geq m_{i-1}$  then
6          $t_{OU} \leftarrow 0$ ;
7          $s \leftarrow \text{Overuse}$ ;
8   else
9      $t_{OU} \leftarrow 0$ ;
10     $s \leftarrow \text{Underuse}$ ;
11 else
12    $t_{OU} \leftarrow 0$ ;
13    $s \leftarrow \text{Normal}$ ;
  
```

根据负载调整带宽

