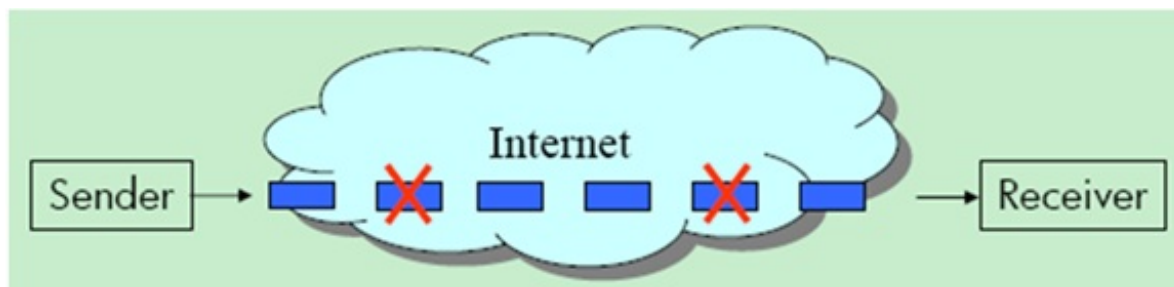


第12章 差错控制

1. 视频传输错误

- 视频传输的信道有可能是不可靠的
 - 无线信道
 - Internet
- 传输错误
 - 随机的比特错误
 - 比特反置、比特插入、比特丢失
 - 突发错误 (Bursty Error)
 - 数据包丢失
 - 局域网数据包冲突
 - 延迟
 - 缓冲区溢出
 - 噪声干扰



2. 差错控制的目的

- 目的
 - 克服由于信道传输错误及有这些错误产生的影响
- 差错控制技术可以应用在
 - 信源编码
 - 信道编码
 - 解码端

3. 差错控制技术

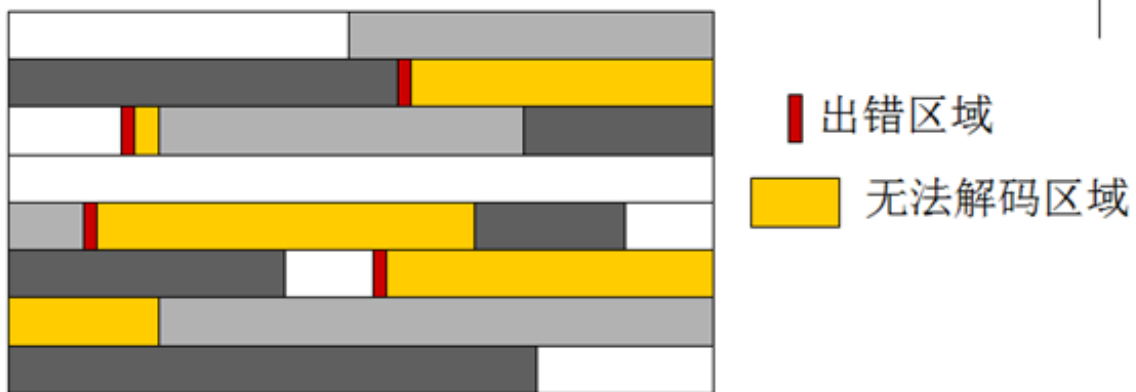
- 数据重传 (Retransmission)
- 前向纠错 (Forward Error Correction , FEC)
- 差错隐藏 (Error Concealment)

- 差错恢复 (Error Resilience)
 - 信道编码
 - 数据重传, FEC
 - 信源编码
 - 差错隐藏, 差错恢复

4. 纠错

- 完美纠错
 - 比特差错检测和纠错
 - FEC
 - 自动重传请求 (Automatic Retransmission Request, ARQ)
- 有损纠错
 - 利用HVS的图像后处理

5. 图像错误



- 条带结构减少空域差错蔓延
 - 条带间解码独立, 差错不能在条带间蔓延
 - 条带内解码相关, 差错能在条带内蔓延
- 帧间预测增加时域差错蔓延
 - 用有差错区域作为参考的编码块不能被解码出正确的图像块

6. 重传

- 条件
 - 编解码器之间的信道为反馈类型信道
- 方法
 - 解码器向编码器反馈信息, 通知编码器重传丢失的数据

- 优点
 - 仅仅需要发送丢失的数据，可以节省带宽
 - 适于变带宽的信道传输
- 缺点
 - 延迟，重传数据需要一定时间，重传次数增加，传输效率下降
 - 不适合单工信道传输，如广播，多播信道

7. 改进重传的算法

- 改进重传算法
 - 有限延迟重传
 - 仅重传能够及时到达的数据
 - 优先级重传
 - 有选择的优先传输重要的数据
 - 视频编码数据的重要程度不同
 - 编码图像类型的重要性
 - I帧：最重要
 - P帧：重要
 - B帧：一般
 - 缩放编码的不同层次的重要性
 - 基本层：最重要
 - 增强层：层次越高，重要性越低

8. 前向纠错码

- FEC
 - 增加冗余数据用于恢复差错



- FEC由两部分组成，前面K字节称为信息码，后面N-K字节称为校验码
- 常用的FEC是Reed-Solomon码
- 应用范围：
 - 存储设备纠错，VCD，DVD
 - 卫星通信，无线通信
 - 数字电视广播

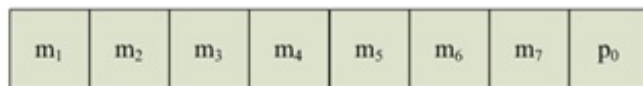
9. 循环冗余码CRC

- 循环冗余码 (n, k) 的编码规则：
 - 将k比特的信息码左移r位 ($r=n-k$)，生成新码字
 - 用一个r比特的校验码生成多项式 $g(x)$ 模2除以新生成的码字，得到的余数作为校验码
 - 校验码附加在信息码的后面构成循环冗余码
- 模2除运算：
 - 在除法计算中采用模2加法运算
 - 模2加法运算就是没有进位的加法计算，即对应位的异或运算
 - $0+0=1+1=0, 1+0=0+1=1$
- 例：校验码生成多项式为 $g(x)=x^4+x^3+x^2+1$ ，计算信息码110的 (7, 3) 循环冗余码
- 110左移4生成新码字1100000
- 校验码生成多项式码字为11101
- 新码字1100000模2除校验码生成码字11101得到校验码1001 (余数)
- 得到循环冗余码
 - $1100000 + 1001 = 1101001$

10. 海明码

$$\begin{array}{r}
 101 \\
 11101 \overline{) 1100000} \\
 \underline{11101} \\
 10100 \\
 \underline{11101} \\
 1001
 \end{array}$$

- 奇偶校验



m_i : 信息码
 p_i : 校验码

- 奇校验：信息码和校验码中“1”的个数为奇数

$$m_1 \oplus m_2 \oplus m_3 \oplus m_4 \oplus m_5 \oplus m_6 \oplus m_7 \oplus p_0 = 1$$

- 偶校验：信息码和校验码中“1”的个数为偶数

$$m_1 \oplus m_2 \oplus m_3 \oplus m_4 \oplus m_5 \oplus m_6 \oplus m_7 \oplus p_0 = 0$$

- 海明距离

- 两个二进制码字间不同比特的数目

$$d(01101, 01100)=1; d(01001, 01100)=2; d(01011, 01100)=3$$

- 如果海明码的海明距离为 $2d+1$ ，可以检测到 $2d$ 比特差错，纠正 d 比特差错

- 海明码编码

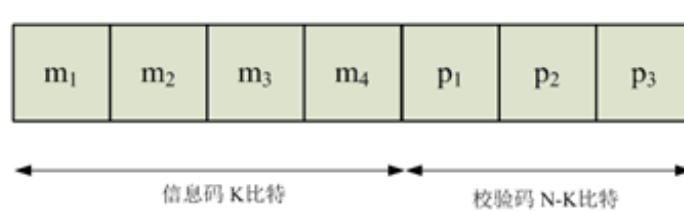
- 编码信息码为海明码须在信息码后添加 r 位校验码并满足 $2^r \geq k+r+1$ 关系

- 以4比特信息码长为例，海明距离为3的校验码的生成方法为

$$\begin{array}{ll}
 m_1 \oplus m_2 \oplus m_3 \oplus p_1 = 0 & m_1 \oplus m_2 \oplus m_3 = p_1 \\
 \text{偶校验} \quad m_1 \oplus m_3 \oplus m_4 \oplus p_2 = 0 & \Rightarrow m_1 \oplus m_3 \oplus m_4 = p_2 \\
 m_2 \oplus m_3 \oplus m_4 \oplus p_3 = 0 & m_2 \oplus m_3 \oplus m_4 = p_3
 \end{array}$$



● 监督关系



p1p2p3	000	100	010	001	011	101	110	111	$m_1 \oplus m_2 \oplus m_3 \oplus p_1 = 0$
差错位置	无错	p1	p2	p3	m4	m2	m1	m3	$m_1 \oplus m_3 \oplus m_4 \oplus p_2 = 0$
									$m_2 \oplus m_3 \oplus m_4 \oplus p_3 = 0$

p1p2p3	000	100	010	001	011	101	110	111	$m_1 \oplus m_2 \oplus m_3 \oplus p_1 = 0$
差错位置	无错	p1	p2	p3	m4	m3	m2	m1	$m_1 \oplus m_2 \oplus m_4 \oplus p_2 = 0$
									$m_1 \oplus m_3 \oplus m_4 \oplus p_3 = 0$

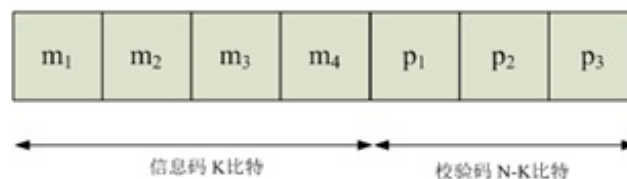
信息码	海明码
0000	0000000
0001	0001011
0010	0010111
0011	0011100
0100	0100101
0101	0101110
0110	0110010
0111	0111001
1000	1000110
1001	1001101
1010	1010001
1011	1011010
1100	1100011
1101	1101000
1110	1110100
1111	1111111

- 最小海明距离 $2d+1=3$
- 有效码率 $4/7$
- 能检测 $2d=2$ 的错误
- 能纠正 $d=1$ 的错误

$$m_1 \oplus m_2 \oplus m_3 = p_1$$

$$m_1 \oplus m_3 \oplus m_4 = p_2$$

$$m_2 \oplus m_3 \oplus m_4 = p_3$$



- 海明码解码
- 解码器收到码字 $\hat{m}_1\hat{m}_2\hat{m}_3\hat{m}_4\hat{p}_1\hat{p}_2\hat{p}_3$

(假设最多出现一比特错误)

s_1	s_2	s_3	差错位置
0	0	0	无错
0	0	1	\hat{p}_3
0	1	0	\hat{p}_2
1	0	0	\hat{p}_1
0	1	1	\hat{m}_4
1	0	1	\hat{m}_2
1	1	0	\hat{m}_1
1	1	1	\hat{m}_3

p1p2p3	000	100	010	001	011	101	110	111
差错位置	无错	p1	p2	p3	m4	m2	m1	m3

$$s_1 = \hat{m}_1 \oplus \hat{m}_2 \oplus \hat{m}_3 \oplus \hat{p}_1$$

$$s_2 = \hat{m}_1 \oplus \hat{m}_3 \oplus \hat{m}_4 \oplus \hat{p}_2$$

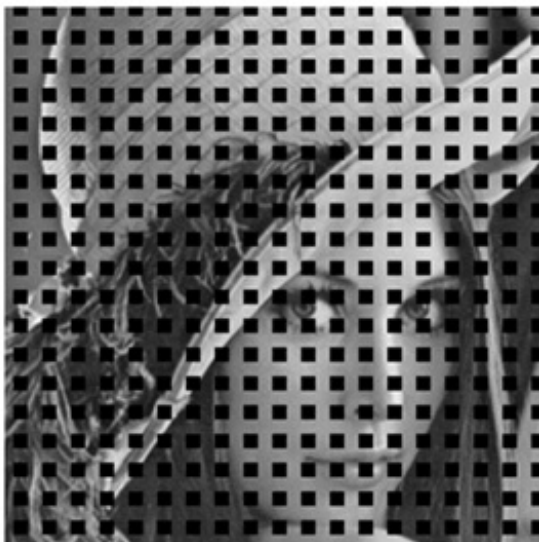
$$s_3 = \hat{m}_2 \oplus \hat{m}_3 \oplus \hat{m}_4 \oplus \hat{p}_3$$

11. 差错隐藏

- 多媒体通信不需要所有数据被精确接收
- HVS对少量的视频错误不敏感
- 一些视频解码差错可以根据HVS的特点被隐藏起来
- 视频编码差错隐藏
 - 估计损失的数据，隐藏损失数据带来的图像差错
 - 在解码端对解码图像做差错隐藏
 - 对编码码流没有影响，因此不增加额外的比特
 - 通常利用图像空域和时域上的相关性做差错隐藏

12. 空域差错隐藏

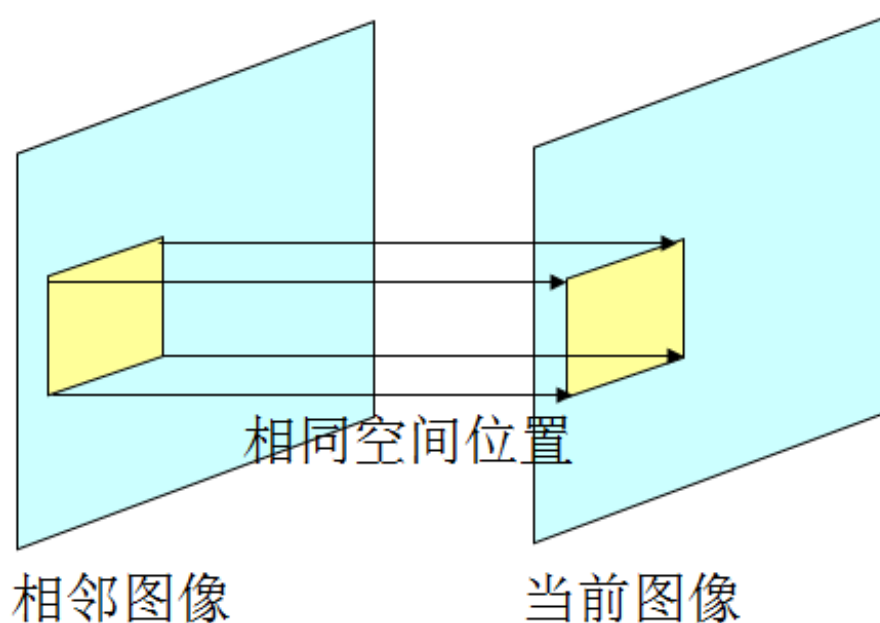
- 空域插值
 - 利用图像内部的信息估计出损失的像素
 - 边缘适应性插值



13. 时域差错隐藏

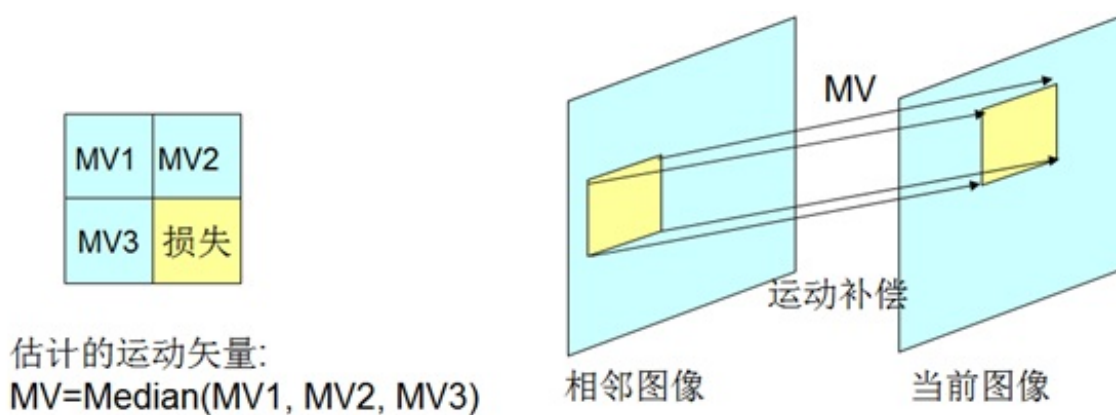
- 时域插值

- 当前图像上损失的像素用其相邻图像上空间位置相同的像素代替
- 缺点：只适合静态场景图像，不适合动态场景图像，降低图像质量



- 运动补偿插值

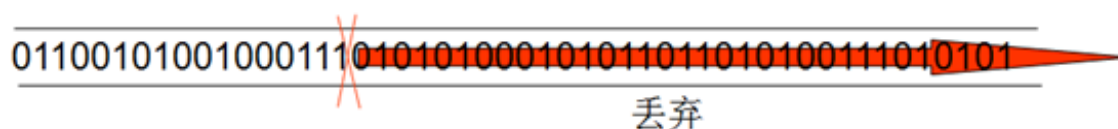
- 当前图像上损失的像素用其相邻图像上运动补偿后的像素代替



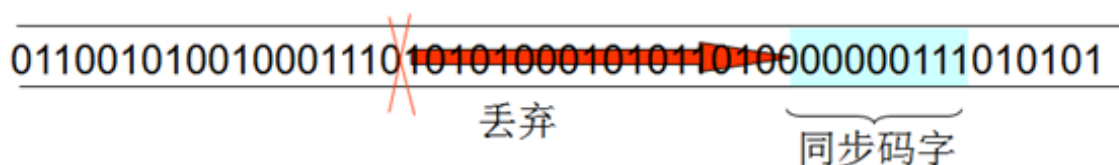
14. 差错恢复

- 在编码端通过编码算法改变码流，增强码流的恢复差错的能力

- 导致差错的两个原因
 - 失去同步
 - 解码器不知道下一个解码的比特属于哪一个语法元素
 - 例如：VLC
 - 解决方法：
 - 跳过出错的数据
 - 重新同步
- 差错传递
 - 编解码端不匹配，导致不同的预测，使差错传递
- 失去同步
 - 码流中任何差错都可能会导致失去同步



- 解决方法：
 - 插入同步码字（起始码）
 - 能够限制差错传递到下一个同步码字
 - 同步码字应该和其它码字区别开
 - 同步码字可以插入到图像，条带的起始位置



15. RVLC

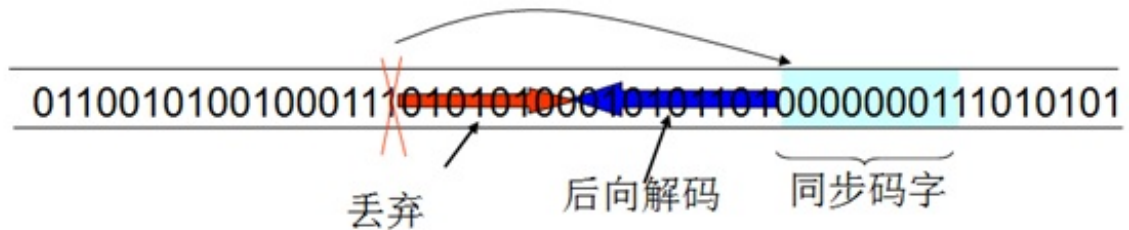
- 传统的VLC仅仅从前向唯一的解码
 - 当一个VLC码字出错，则从当前解码码字开始到下一个同步码字之间的码字不能解码都要丢掉
- RVLC既可以前向解码，也可以后向解码
 - 如果一个RVLC码字出错，跳到下一个同步码字前，从后向解码，部分恢复数据
 - MPEG-4，H.263+标准支持RVLC的差错恢复

- RVLC

前缀码的第一个和最后一个比特设为"1"

前缀码的偶比特位设为"0"

前缀码的奇比特位用来识别不同的码字



16. 数据分区 (Data Partitioning)

- 将数据分成重要数据和一般数据分别放在码流的不同位置
 - 重要的数据紧跟着放在同步字的后边
 - 重要数据：编码模式，MV，DC系数
 - 一般数据放在重要数据的后边
 - 一般数据：AC系数
- 重要数据先于一般数据解码

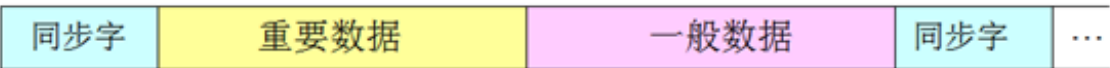
	Non-Reversible Exp-Golomb Code with k=1		Reversible Exp-Golomb Code with k=1	
	Prefix	Suffix	Prefix	Suffix
0	0	0	0	0
1	0	1	0	1
2	100	0	1 0 1	0
3	100	1	1 0 1	1
4	101	0	1 1 1	0
5	101	1	1 1 1	1
6	11000	0	1 0 0 0 1	0
7	11000	1	1 0 0 0 1	1
8	11001	0	1 0 0 1 1	0
9	11001	1	1 0 0 1 1	1
10	11010	0	1 1 0 0 1	0
11	11010	1	1 1 0 0 1	1
etc.				

17. 差错传递问题

- 编解码器不匹配产生不正确的预测，导致差错传递
 - 例如：运动补偿预测，DC预测

18. 差错传递限制

- 周期的插入I帧
 - 差错可以被I帧阻止
 - 降低编码效率
- 在空间预测帧中插入I宏块



- 可以在空间预测帧中差错传递较大的地方采用Intra宏块编码阻止差错进一步传递
- 编码效率比I帧高

19. 反馈编码

- 如果编解码之间的信道为反馈信道，可以根据解码器的反馈信息重新预测编码

- 编码器重新编码出错帧为I帧，重传给解码器

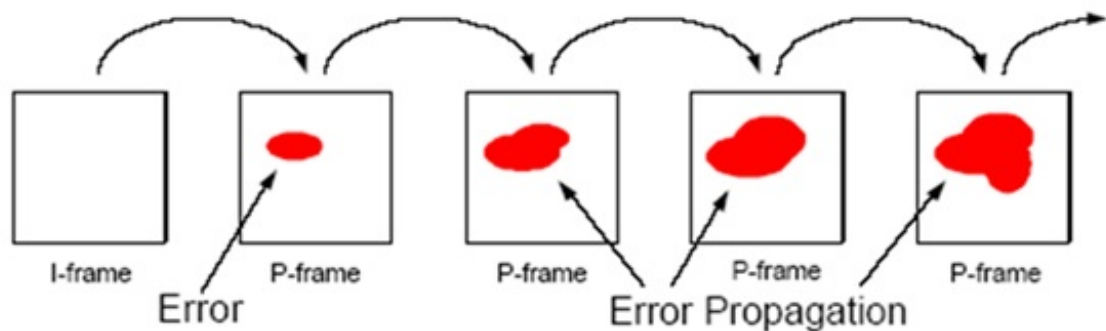
- 比周期插入I帧更简单，能及时避免误差传递
- 编码效率低

- 编码器用其它正确的参考帧做预测

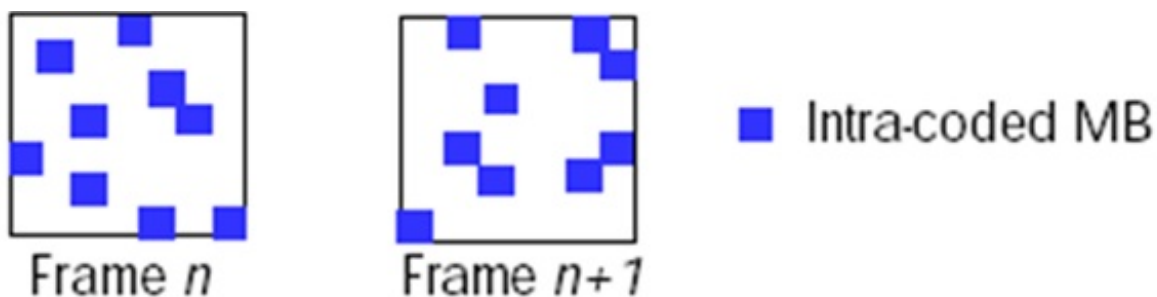
- 编码器和解码器都缓存有多个以前解码的图像
- 编码器在多个以前解码的图像中选择正确的作为参考帧来作为当前图像的参考帧

- 缺点：需要高速的信道传输，否则产生延迟

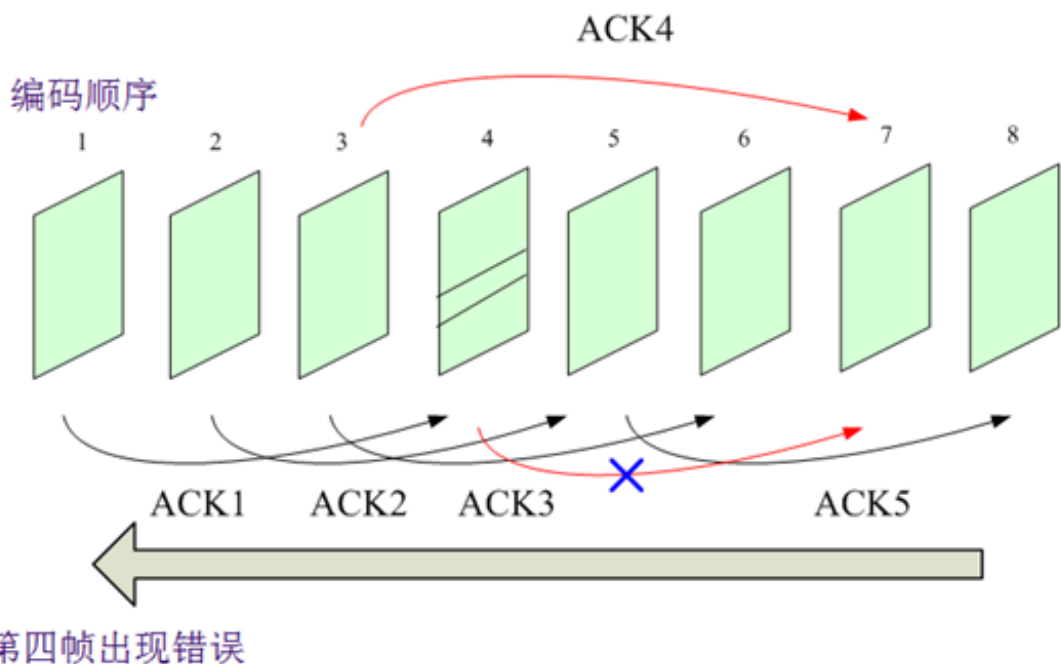
20. 多描述编码 (Multiple Description)



运动补偿差错传递



在空间预测帧中插入Intra编码块

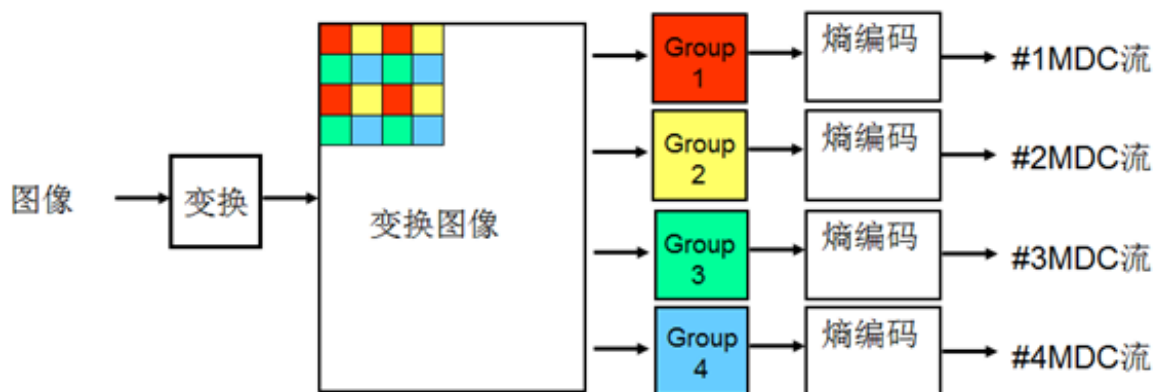
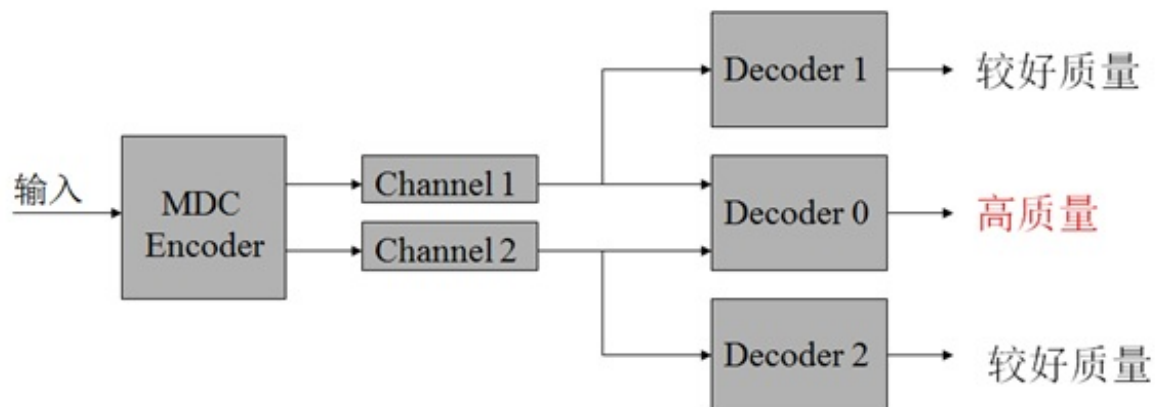


Coding , MDC)

- 视频输入到并行的信道，划分成几部分，经过同样的编码，产生不同的码流，有同样的编码质量
- 图像可以用任意一个码流恢复
- 多个码流合并可以得到更高的图像质量

21. 多描述编码技术

- 空域下采样
- 变换域下采样
- 多描述标量量化



(Multimedia Description Scalar Quantizer , MDSQ)

22. 传输层差错控制

- 鲁棒性打包
 - 编码模式被重复的放入连续的数据包中
- 空间块交织打包
 - 空间连续的块被放如不连续的数据包中
- 加倍传输重要信息
 - 图像类型，编码模式，运动矢量

23. 交互差错控制

- 选择编码
 - 避免用差错区域作为预测编码
 - 对于高误码率的传输，用更多的Intra宏块编码和更短的条带编码

- 无等待重传

- 像素即使出现差错，仍继续解码，记录差错像素位置
- 直到重传数据到达，纠正出错的像素
- 能无延迟的完美重构出解码图像

- 多重复重传

- 对于误码率高的传输信道，多次重复的传输数据



(a) 4个MDC流



(b) 3个MDC流



(c) 2个MDC流



(d) 1个MDC流

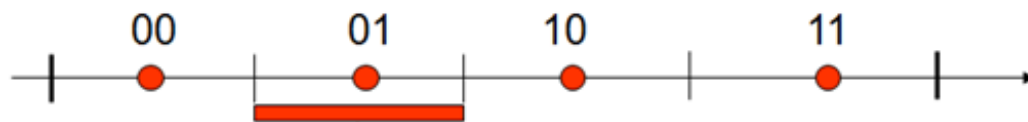
第13章 视频传输

1. 多媒体传输

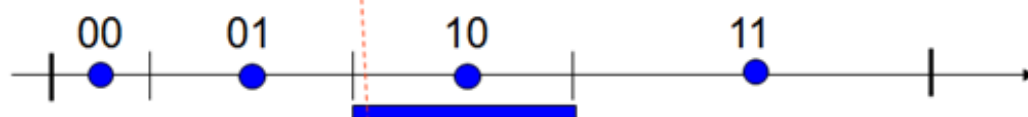
- 多媒体数据通过信道从一端到另一端的通信

2. 多媒体传输特点

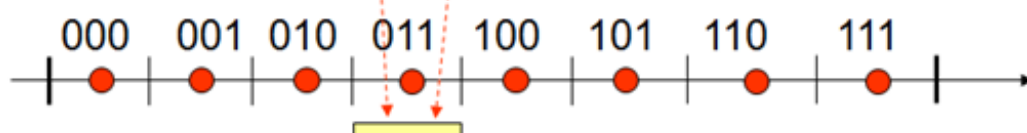
量化器 1



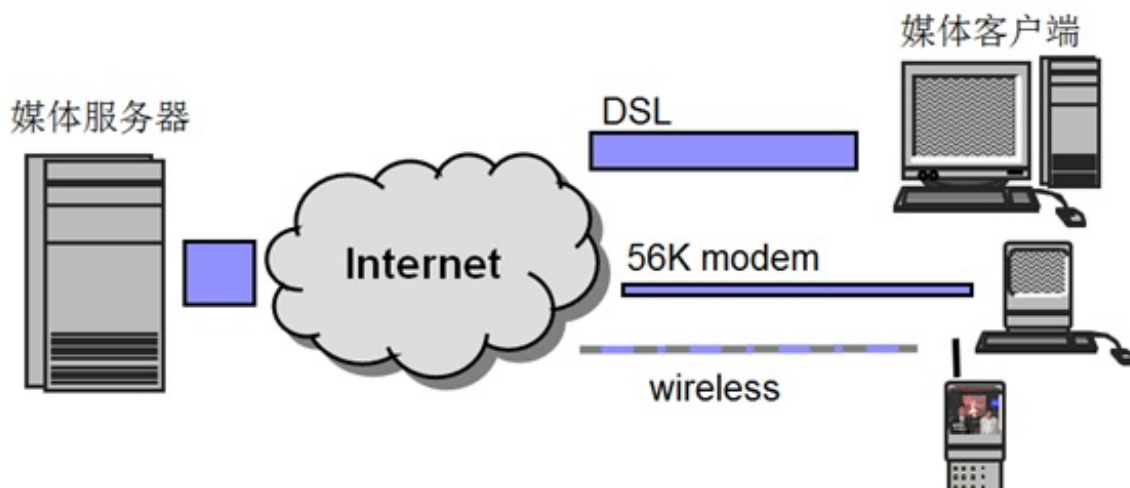
量化器 2



联合量化器 0



- 延迟敏感
(Delay)



Sensitive)

- 端到端延迟
- 抖动延迟
- 数据损失容忍 (Loss Tolerant)
 - 少量的多媒体数据损失是可以接受的
- HVS对多媒体数据的延迟比数据损失更加敏感
- 多媒体数据与一般数据不同
 - 一般数据不允许有损失，但可以有延迟

3. 多媒体传输存在的问题

- 带宽：
 - 有限的带宽
 - 时变的带宽
- 延迟：
 - 解码器得不到解码需要的数据
- 传输数据损失
 - 压缩数据对数据损失非常敏感
 - 可以通过差错控制技术来减少传输数据损失

4. 多媒体传输应用

- 存储多媒体数据传输
 - VCD，DVD
- 实时多媒体数据传输
 - 数字电视广播，电视会议，视频监控
- 流媒体
 - IPTV，网络视频点播

5. 多媒体传输考虑的因素

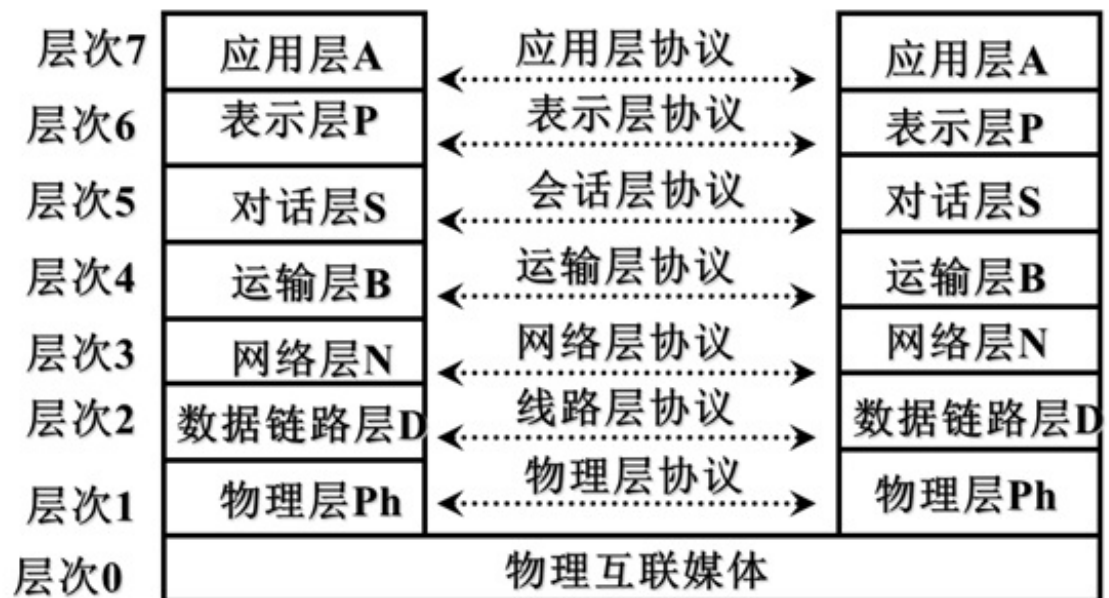
- 差错控制
- 服务质量 (Quality of Service , QoS)
 - 延迟
 - 带宽
 - 丢包率
 - 误码率
- 实时性
- 同步
 - 视频，音频同步
- 实现代价

6. 网络七层协议

7. 网络通信

• 层次结构

- 不同通信任务安排在不同的层次上完成
- 层与层之间通过很好的接口定义来通信
- 层次结构利于通信协议的设计和实现
- 层次越高，逻辑上越接近用户
- 层次越低，越接近物理信号传输



ISO/OSI 参考模型

8. 电路交换网 (Circuit-Switched Network)

- 电路交换网
 - 在一个物理信道可以有几个时分复用的数据连接
 - 在一个数据连接工作期间，一个专门用于该数据连接的电路被建立

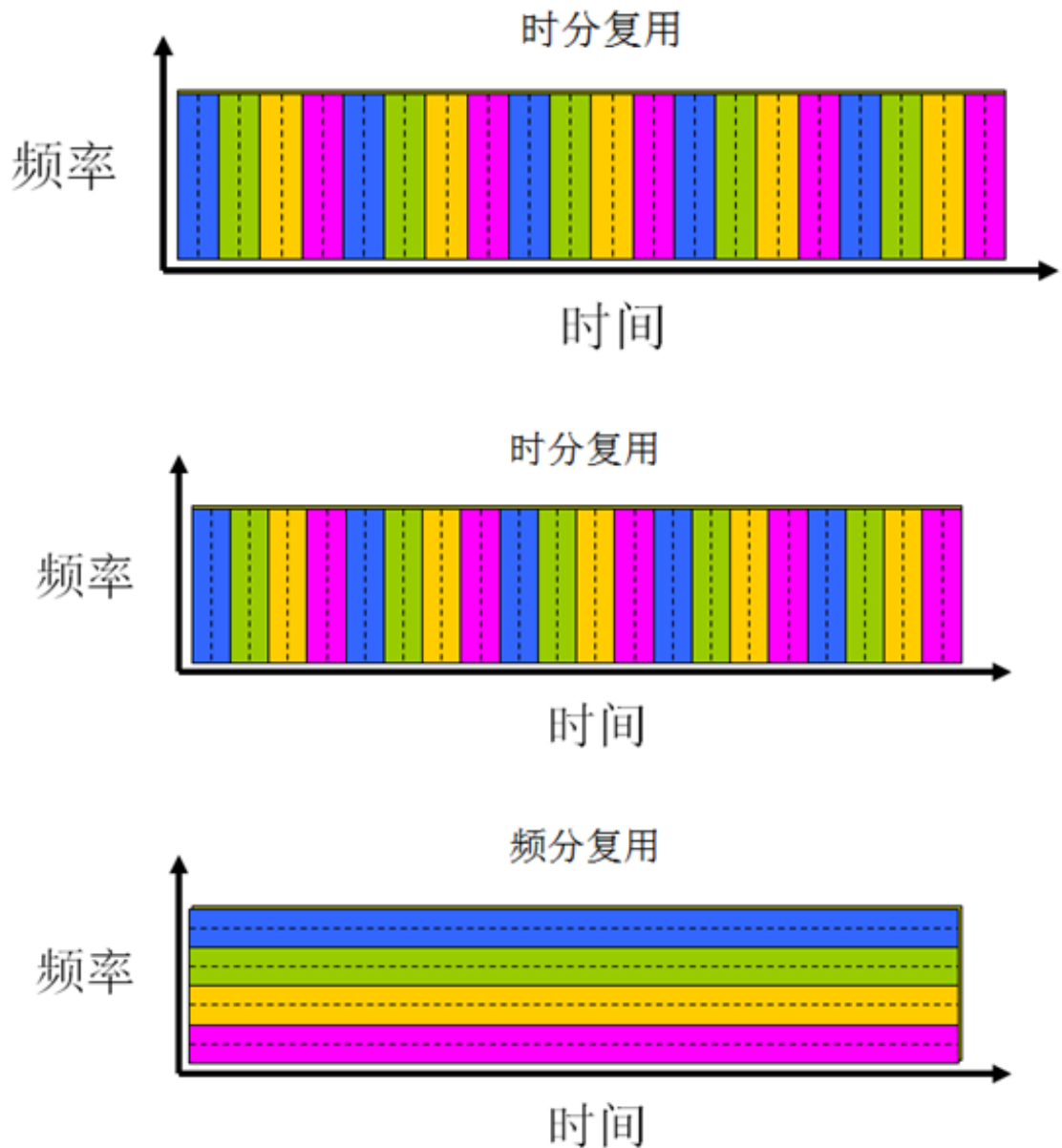
9. 信号复用

- 电路交换网的特点

- 带宽不变
- 传输延迟短
- 小的延迟抖动

- 电路交换网使用案例

- 公共交换电话网



(Public Switch Telephone Network , PSTN)

- 综合业务数字网 (Integer Service Digital Network , ISDN)

- 优点

- 有固定带宽，适合实时应用
 - 音频流
 - CBR视频

- 缺点

- 不适合突发的应用
 - 文件传输，电子邮件，网页浏览
 - VBR视频

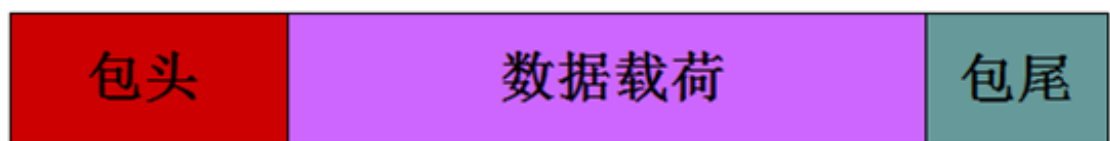
20. 包交换网 (Packet-Switched Network)

- 数据以包的形式在信道中传输

- 数据包
 - 包头
 - 包ID，包长度，IP地址，同步字，传输协议等
 - 数据载荷
 - 打包的有效数据
 - 包尾
 - 冗余信息或校验信息

- 无连接的
- 丢失可重传

- 包交换网的特点



- 通信连接给多个用户共享，来自不同用户的数据包共享同一个网络资源
- 允许更多的用户使用网络
- 每个数据包可以占用信道的整个带宽
- 如果信道繁忙，则新的数据包将被放到缓冲区队列中
- 如果缓冲区队列充满，则数据包丢失
- 包交换网的特点
 - 数据包长度可变
 - 大的传输延迟
 - 大的延迟抖动
- 包交换网使用案例
 - 局域网
 - 以太网：IEEE 802.3
 - 令牌环：IEEE 802.5
 - 广域网
 - Internet
 - 移动网
 - 通用无线分组业务（General Packet Radio Service，GPRS）

- 优点
 - 适合需要动态带宽的应用
 - 数据
 - VBR视频
- 缺点
 - 不适合实时的应用
 - 实时音视频传输

21. 电路交换与包交换

- 连接带宽1Mb/s
- 每个用户占用带宽100kb/s
- 电路交换网最大只能有10个用户
- 包交换网
 - 如果大于10用户同时使用网络的概率小于0.004
 - 可以支持35个用户使用，服务质量和电路交换网一样好

22. 包延迟

- 处理延迟
 - 数据校验，纠错
 - 头部检查
- 排队延迟
 - 传输前在队列中的等待时间
- 转发延迟
 - 发送比特到信道的时间
- 传输延迟
 - 在信道中传输的时间

23. 视频传输协议

- 网络传输协议
 - TCP
 - UDP
 - RTP/RTCP：运行在UDP上的传输层协议
- 视频传输协议基于网络传输协议
 - 采用RTP/RTCP协议传输视频

24. TCP和UDP

- TCP：传输控制协议（Transmission Control Protocol）
 - 每个数据包的传输都需要应答
 - 提供可靠的数据包传输
 - 长延迟
 - 面向连接的协议
- UDP：用户数据报协议（User Datagram Protocol）
 - 数据包传输无应答
 - 提供尽量好的数据传输
 - 无连接的简单协议

25. RTP

- RTP：实时传输协议（Real-time Transport Protocol）
 - 在UDP上运行的实时传输协议
 - 时间戳机制解决延迟抖动
 - 数据包序列号机制为接受的数据包排序
 - 数据载荷类型信息识别数据载荷的格式
 - H.261，MPEG-1，H.262（MPEG-2）

26. RTCP

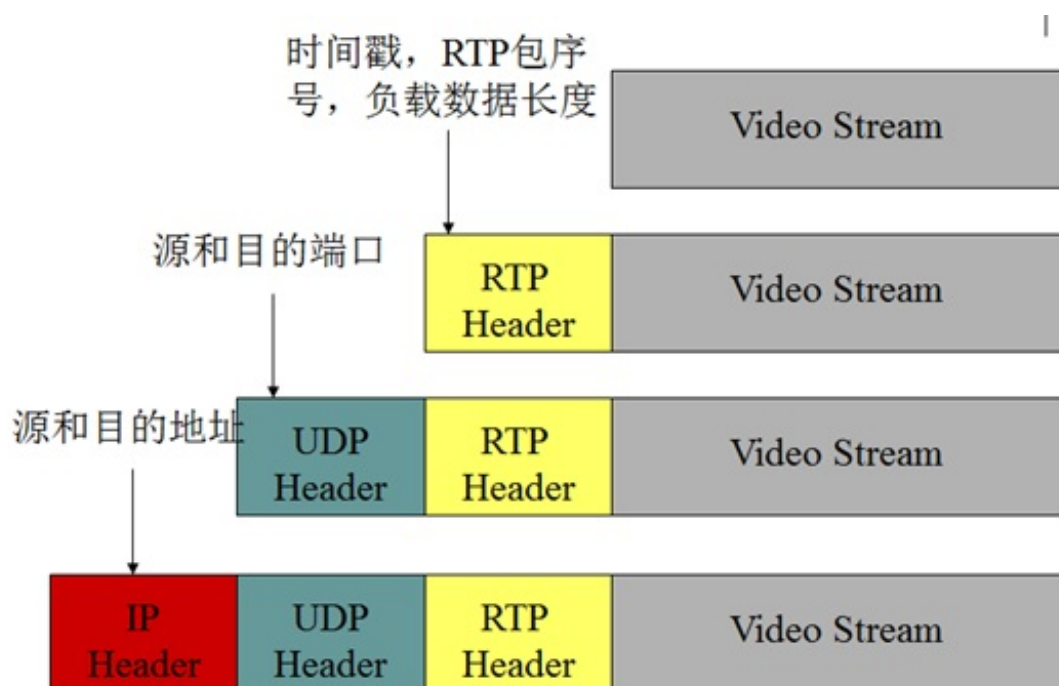
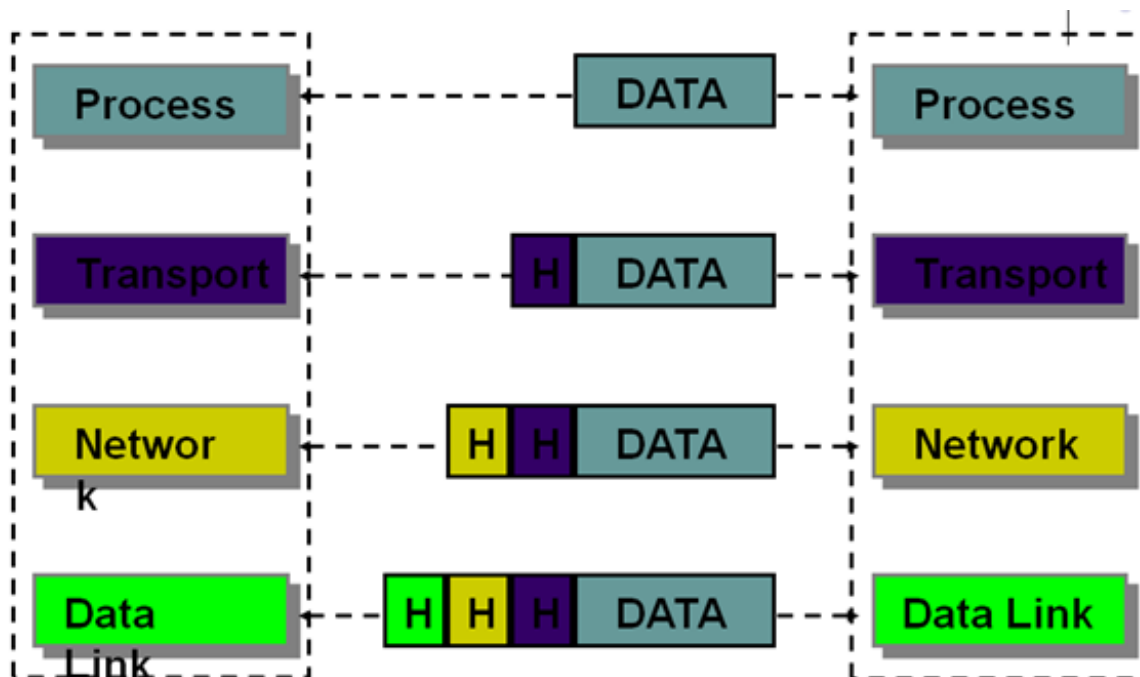
- RTCP：实时传输控制协议（Real-time Transport Control Protocol）
 - 与RTP配套使用的控制协议
 - 用于监视网络的QoS并传输一些上层信息
 - 发送和接收监控信息，例如：往返传输延迟，数据包丢失率，数据到达抖动，

27. 视频数据封装

28. 视频的网络传输

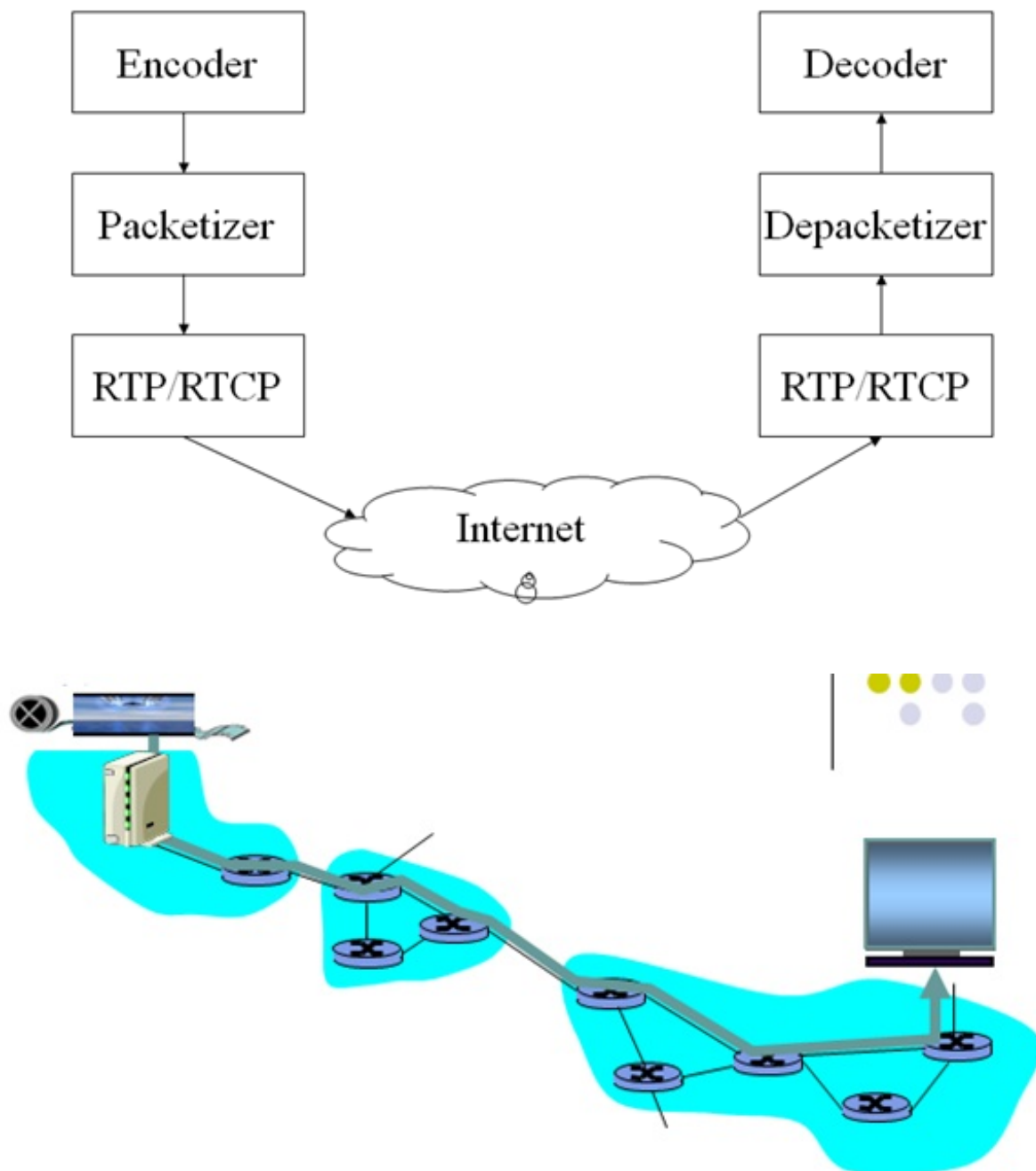
29. 流媒体

- 流媒体存储在服务器端，通过信道传输到客户端
- 流媒体特点
 - 连续实时传输
 - 不必等到流媒体文件全部下载完毕才传输，用户只需等待几秒，十几秒的启动延时即可观看，启动延时大大减少
 - 流媒体文件传输和观看同时进行
 - 不需要客户端有很大的存储空间



- 流媒体交互功能

- 有暂停、回放、快进、快退、拖动等功能
- 只需十几秒的初始延迟
- 操作延时只有1, 2秒钟
- 客户端的回放缓冲



据，提供快速回放的功能

- 长时间回放则需要数据重传

- 例子：

- 网络直播电台
- 体育赛事网络直播

- 和一般流媒体一样在客户端有回放缓冲

- 交互：

- 无快进功能
- 可以回放，暂停

- 带宽

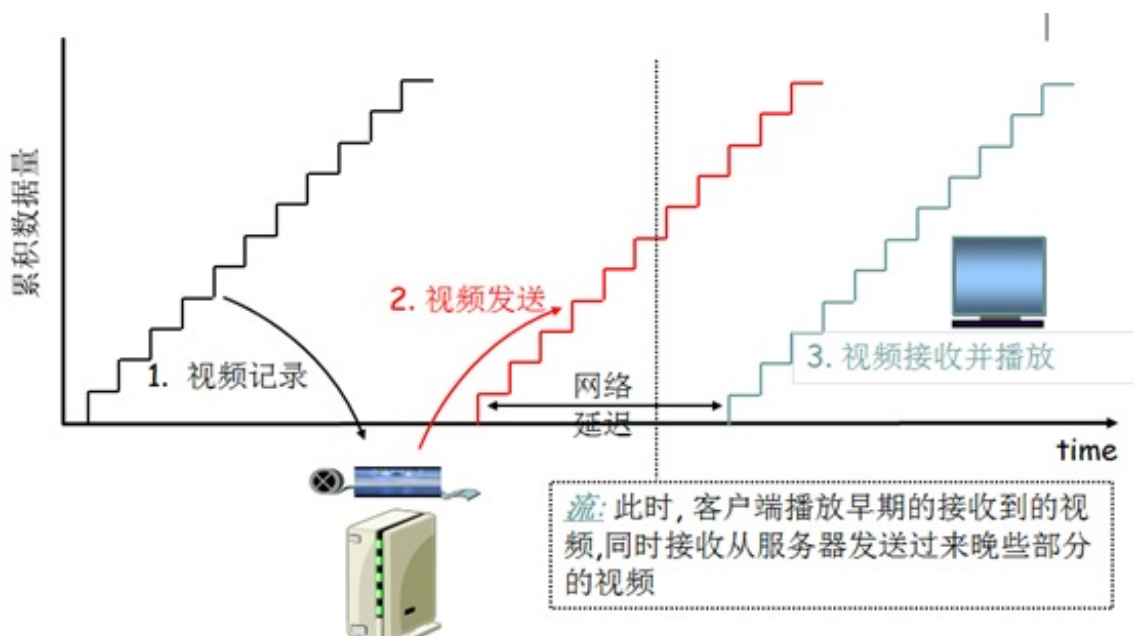
- 网络带宽是时变的
- 需要控制码流适应带宽

- 端到端延迟

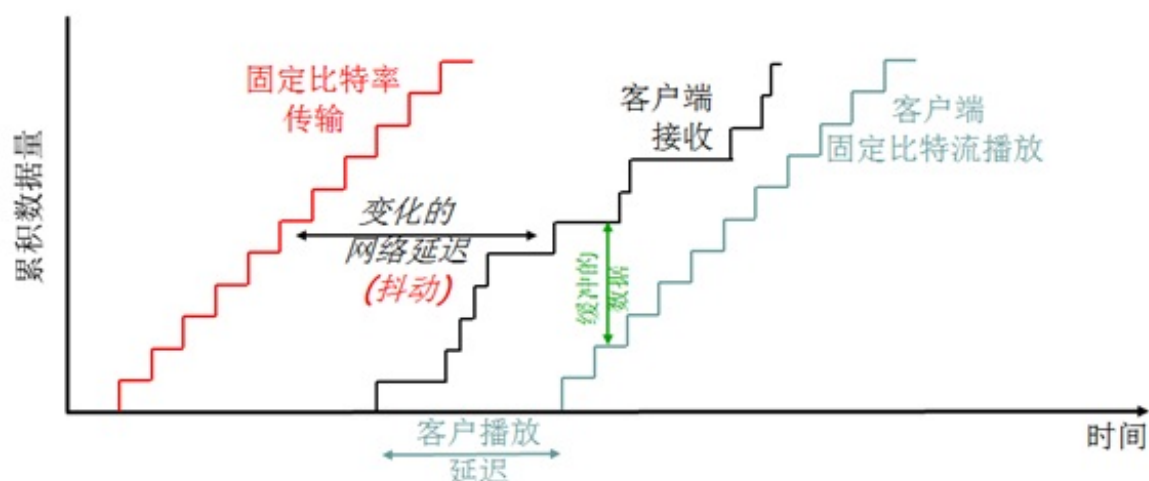
- 用缓冲区控制处理延迟和延迟抖动

- 传输数据丢失

- 压缩的码流对数据丢失非常敏感
- 需要错误控制来恢复丢失的数据



流媒体延迟抖动



31. 多媒体系统结构

- 应用层

- 原始的未压缩的视频、音频

- 压缩层

- 压缩的视频基本流，音频基本流

- 传送层

- 将基本流打包成基本流数据包，并实现音视频的基本流数据包复用，组成复用的传送流

- 传输层

- 信道编码，载波调制产生信号发送到信道中并传输

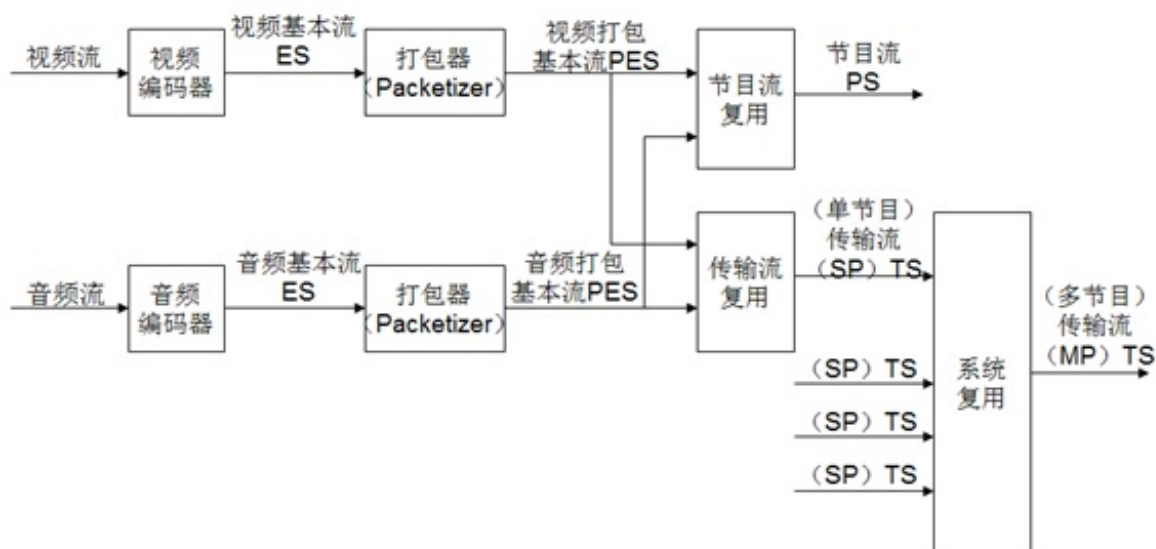
32. 音频复用

- 音频和视频码流通过复用实现同步

33. 码率控制

- 码率

- 不同是视频图像有不同的编码比特数



- CBR

- 每帧图像采用相同的比特数编码，码率固定
- 每帧图像的质量 (PSNR) 不固定
- ISDN，DTV

- VBR

- 每帧图像采用不同的比特数编码
- 每帧图像的质量固定
- DVD，无线网络，Internet

- 视频存储编码的码率控制

- 编码固定时间的视频到固定比特数为R的码流
- 例如：DVD，4.7GB的空间存储两小时的电影

- 限定比特数的码率控制方法

- 分配相等的比特数到每一帧， R/N bit/frame
 - 存在质量不稳定的问题
- 多趟编码
 - 视频存储编码没有实时的要求
- 可以反复检查整个视频序列并且重新编码
- 能提供比单趟编码更好的编码码率控制性能

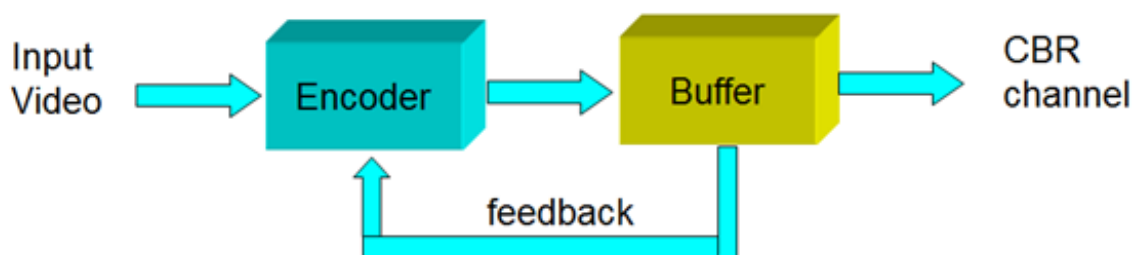
- 视频码率可以通过以下方法控制

- 调整量化参数 (最常用)
- 调整帧率
- 调整空间分辨率
- 增加或丢弃增强层 (用于可分级编码)

- 存储视频的多趟编码方法
 - 采用VBR编码整个视频序列
 - 收集并分析编码的统计信息
 - 图像质量及编码比特数
 - 如果所有编码比特数大于限定的最大比特数
 - 找到视频序列的复杂部分
 - 从新为复杂的图像分配比特数
 - 从新编码整个视频序列
- 数字电视信道
 - 20Mb/s带宽
 - CBR
 - 通过对缓冲区情况的反馈来控制产生比特率
 - 如果缓冲区水平太高，增加量化步长
 - 如果缓冲区水平太低，减少量化步长

34. 缓冲区管理

- 编解码缓冲区

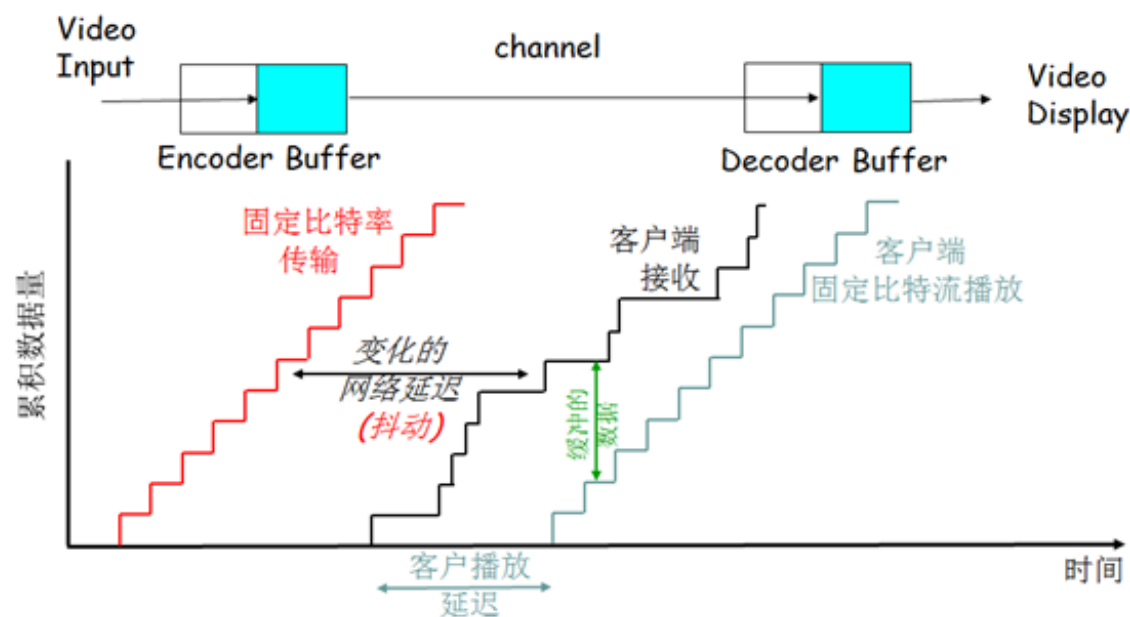


- 使用缓冲区的好处

- 减少抖动
- 通过重传进行差错恢复
- 通过交织编码进行差错恢复
- 平滑输入输出波动

- 缓冲区问题

- 上溢
- 下溢

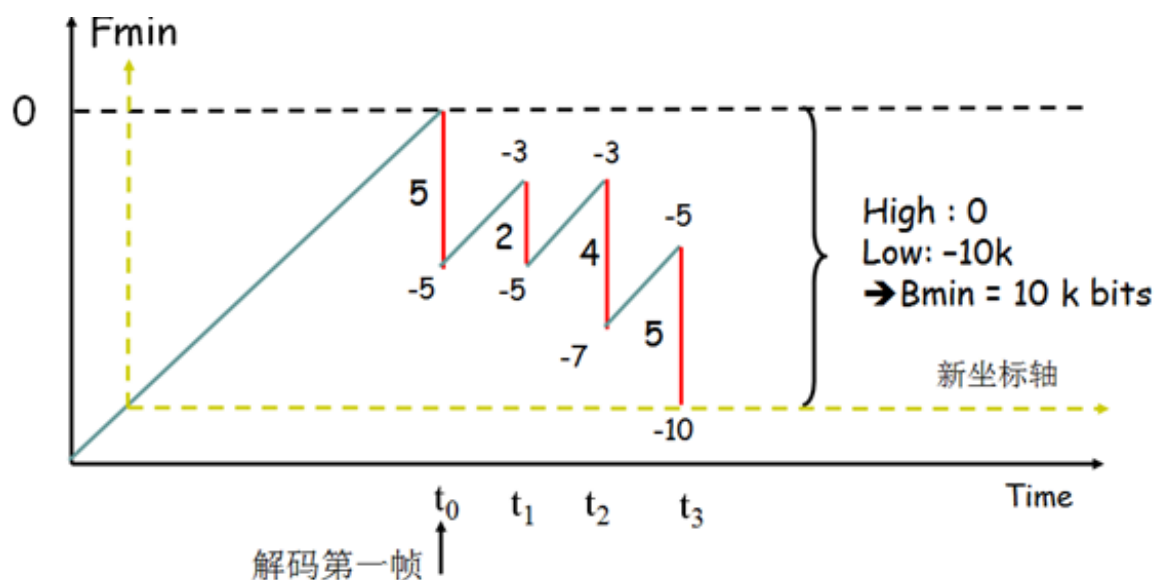


- 假设参考模型 (Hypothetical Reference Decoder, HRD)
- 用于H.263, MPEG, H.264

- 漏斗参数
 - R : 峰值传输码率
 - B : 缓冲区大小
 - F : 初始解码缓冲区的充满度
- HRD的使用
 - 编码器编码完一段视频序列后能发现一个 (R, B, F) 参数三元组, 将其发送给解码器, 使解码器使用这些参数有效的解码该视频序列
 - 解码器可以根据这些参数确定是否能解码一个码流, 并且知道初始的延时是多少
 - HRD算法: 给定压缩码流 $\{b_0, b_1, \dots, b_{N-1}\}$ 和码率R, 发现最小的缓冲区大小 B_{min} 和最小的初始缓冲区的充满度 F_{min}
 - 解码压缩码流, 不考虑缓冲区的上溢和下溢, 找到缓冲区存储码流的最高层High和最底层Low
 - $B_{min} = High - Low$, $F_{min} = -Low$
- HRD的例子: 帧率30fps, 序列编码比特 $b_i = \{5k, 2k, 4k, 5k\}$, 传输码率 $R = 60\text{kbps} \Rightarrow 2\text{kb/frame}$

- HRD的例子: 解码器缓冲区的解码状态

- 初始延迟:



$$10\text{kb}/60\text{kbps} = 1/6 \text{秒}$$

