



计算机网络概论

应用层

刘志敏

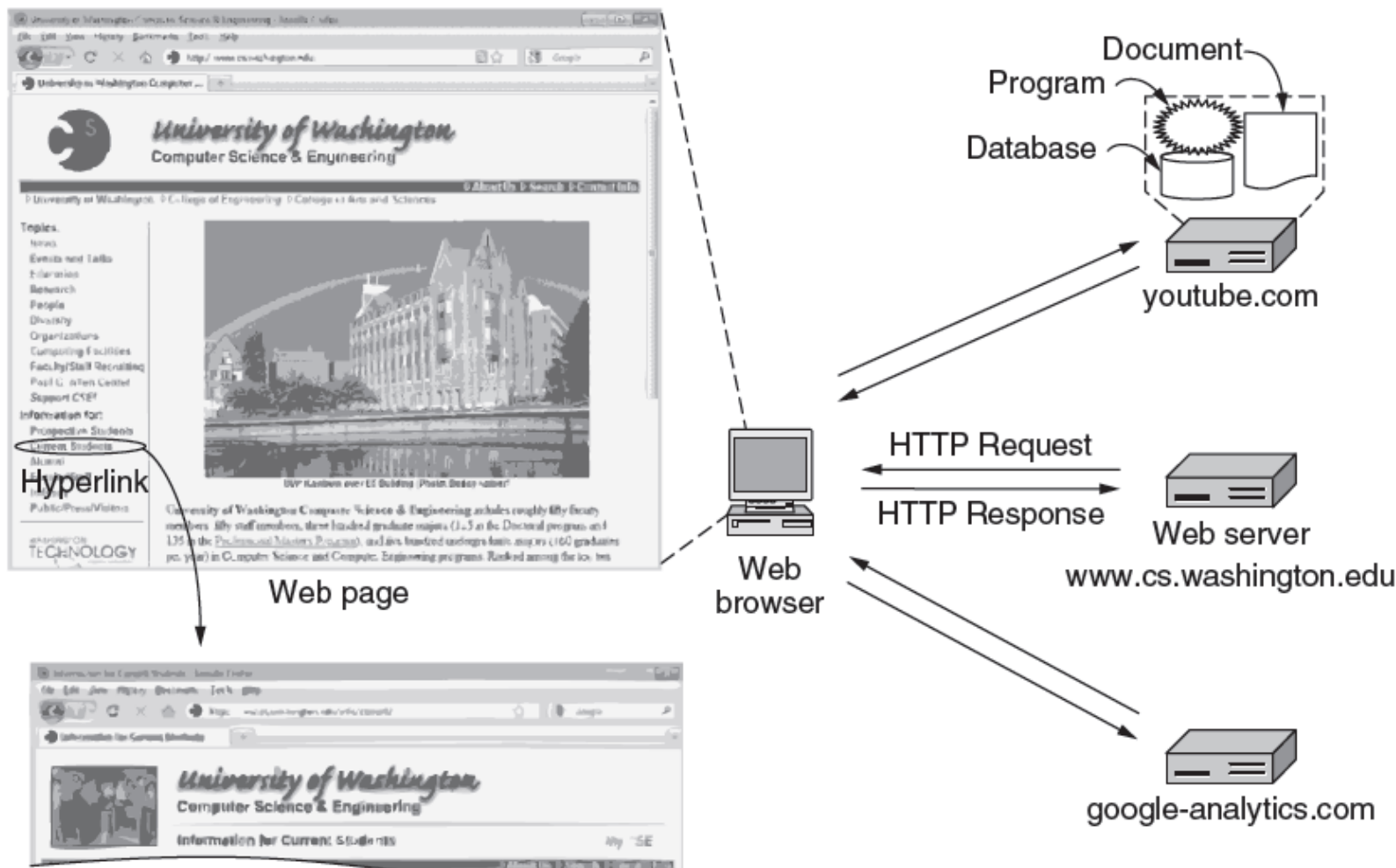
liuzm@pku.edu.cn



提纲

- Web服务（通过阅读教材的第7章）
 - 网页、URL
 - 访问Web服务的过程：浏览器、服务器、HTTP
 - 页面缓存、Cookies作用
- 流媒体应用与协议
 - 数字音频与数字视频
 - 互联网流媒体协议
 - RTP/RTCP, RTSP
 - 媒体播放器

Web 体系结构概述(1)



- Web页面（HTML，超文本、超链接），资源描述URL（统一资源定位符），浏览器与服务器之间的数据传输采用HTTP



Web 体系结构概述(2)

建立Web时，为了让一个页面指向另一个Web页面，需要命名并定位页面，首先需要回答三个问题：

1. 这个页面叫什么？URL
2. 这个页面在哪里？DNS→IP
3. 如何访问这个页面？HTTP(用TCP承载)



Web 体系结构概述(3)

■ 用户选择链接时，客户端（浏览器）的动作：

1. 确定URL
2. 请求DNS解析IP地址，得到DNS响应
3. 建立TCP连接
4. 发送HTTP请求页面
5. 得到服务器的HTTP响应
6. 提取其他的URL
7. 显示页面/index.html
8. 若短时间没有向同一服务器发出请求，则释放TCP连接

■ 服务器端的动作

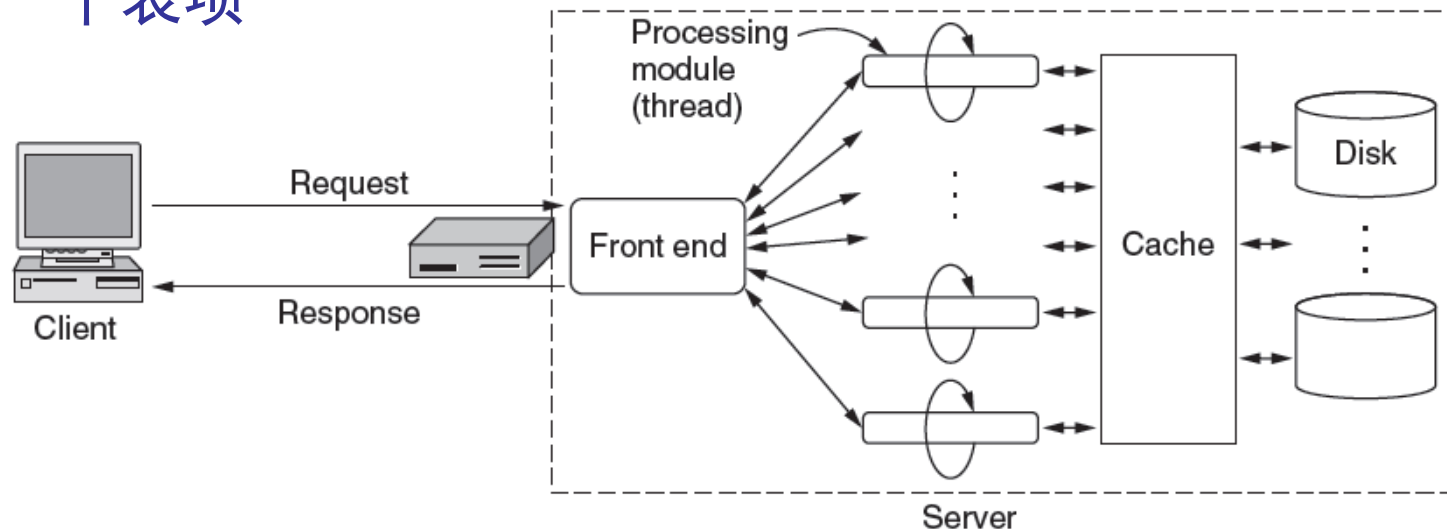
1. 接受客户端的TCP连接
2. 获取页面路径，即文件名
3. 从磁盘上读取文件
4. 将文件内容发送给客户
5. 释放TCP连接

Web 体系结构概述(4)

如何使单位时间内处理的请求更多？

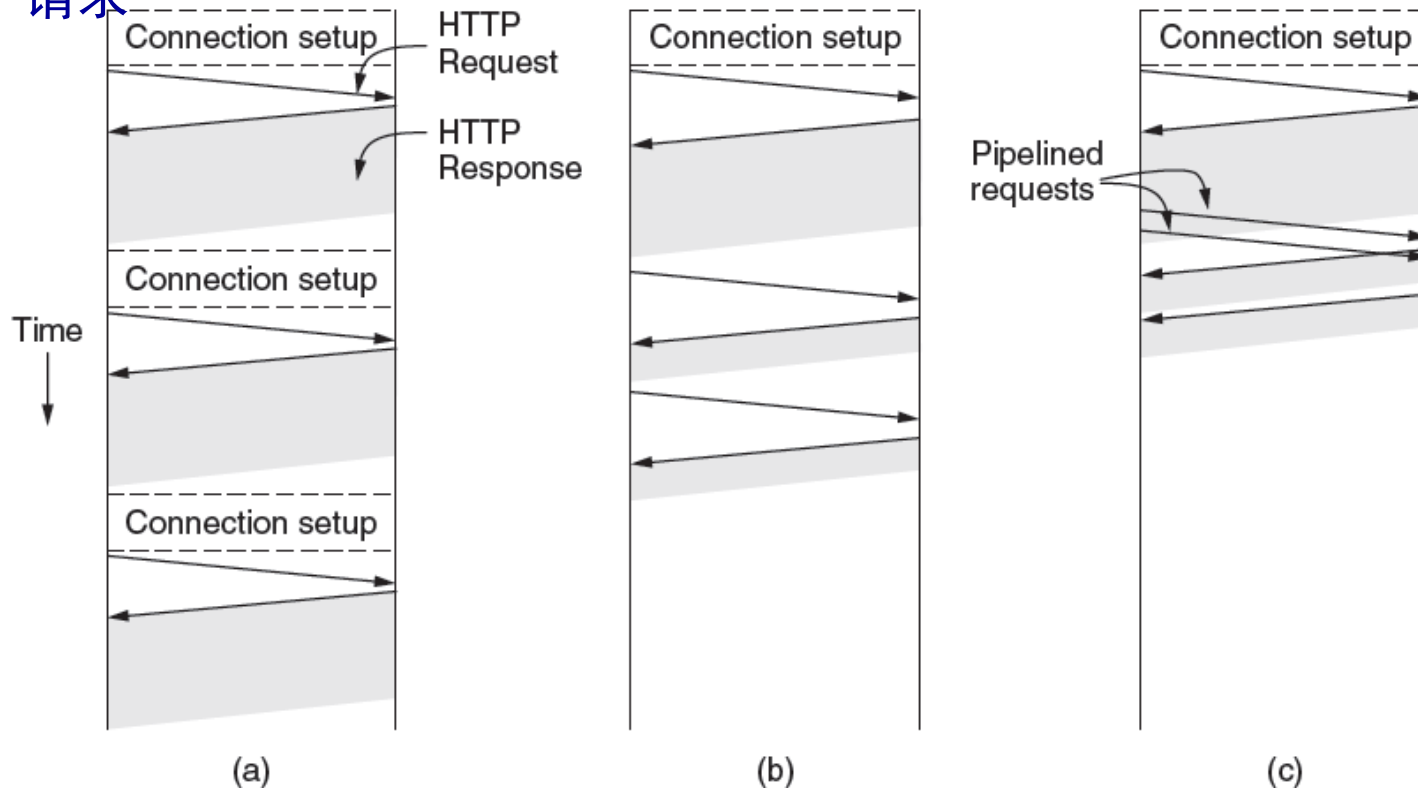
Web服务器的方案：一个前端和多个多线程的处理模块
(处理模块与前端模块共享地址空间；优先在缓存中检索页面文件，提高访问速度)

处理模块的动作：解析Web页面的名字；执行对该页面的访问，检查缓存；从磁盘上获取请求页面或创建页面的程序；将响应返回给客户；在服务器日志中增加一个表项



HyperText Transfer Protocol (1)

- (a) 多个请求多个连接：HTTP1.0，为每个请求都建立1个连接
- (b) 一个持续连接和系列请求：HTTP1.1 连接重用，减少连接建立时间，TCP传输更快，没有TCP的慢启动问题；HTTP响应之后再发出请求
- (c) 一个持续连接和流水线式请求：HTTP1.1 连接重用，可发送多个请求





HyperText Transfer Protocol (2)

Method	Description
GET	Read a Web page
HEAD	Read a Web page's header
POST	Append to a Web page
PUT	Store a Web page
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Connect through a proxy
OPTIONS	Query options for a page

例如
请求页面
请求消息头，测试URL

内置的HTTP请求方法（区分大小写），例如：

GET filename HTTP/1.1



HyperText Transfer Protocol (3)

响应组的状态码

Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later



HyperText Transfer Protocol (4)

某些HTTP 消息头

Header	Type	Contents
User-Agent	Request	Information about the browser and its platform
Accept	Request	The type of pages the client can handle
Accept-Charset	Request	The character sets that are acceptable to the client
Accept-Encoding	Request	The page encodings the client can handle
Accept-Language	Request	The natural languages the client can handle
If-Modified-Since	Request	Time and date to check freshness
If-None-Match	Request	Previously sent tags to check freshness
Host	Request	The server's DNS name
Authorization	Request	A list of the client's credentials
Referer	Request	The previous URL from which the request came
Cookie	Request	Previously set cookie sent back to the server
Set-Cookie	Response	Cookie for the client to store
Server	Response	Information about the server



HyperText Transfer Protocol (5)

某些HTTP 消息头

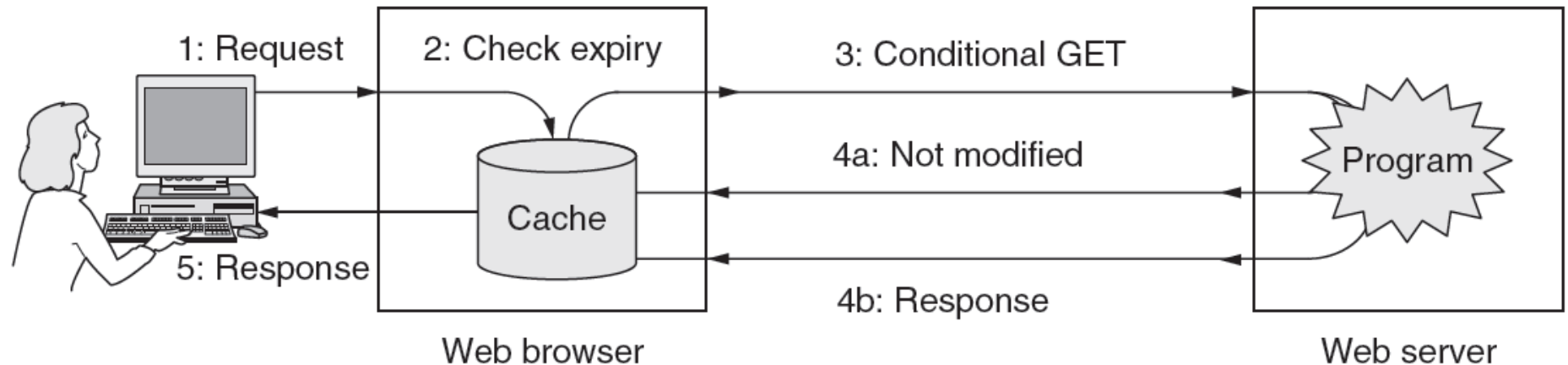
Content-Encoding	Response	How the content is encoded (e.g., <i>gzip</i>)
Content-Language	Response	The natural language used in the page
Content-Length	Response	The page's length in bytes
Content-Type	Response	The page's MIME type
Content-Range	Response	Identifies a portion of the page's content
Last-Modified	Response	Time and date the page was last changed
Expires	Response	Time and date when the page stops being valid
Location	Response	Tells the client where to send its request
Accept-Ranges	Response	Indicates the server will accept byte range requests
Date	Both	Date and time the message was sent
Range	Both	Identifies a portion of a page
Cache-Control	Both	Directives for how to treat caches
ETag	Both	Tag for the contents of the page
Upgrade	Both	The protocol the sender wants to switch to

HyperText Transfer Protocol (6)

HTTP缓存：保留已获取的网页供日后处理，避免重复传输，减少流量和延迟。

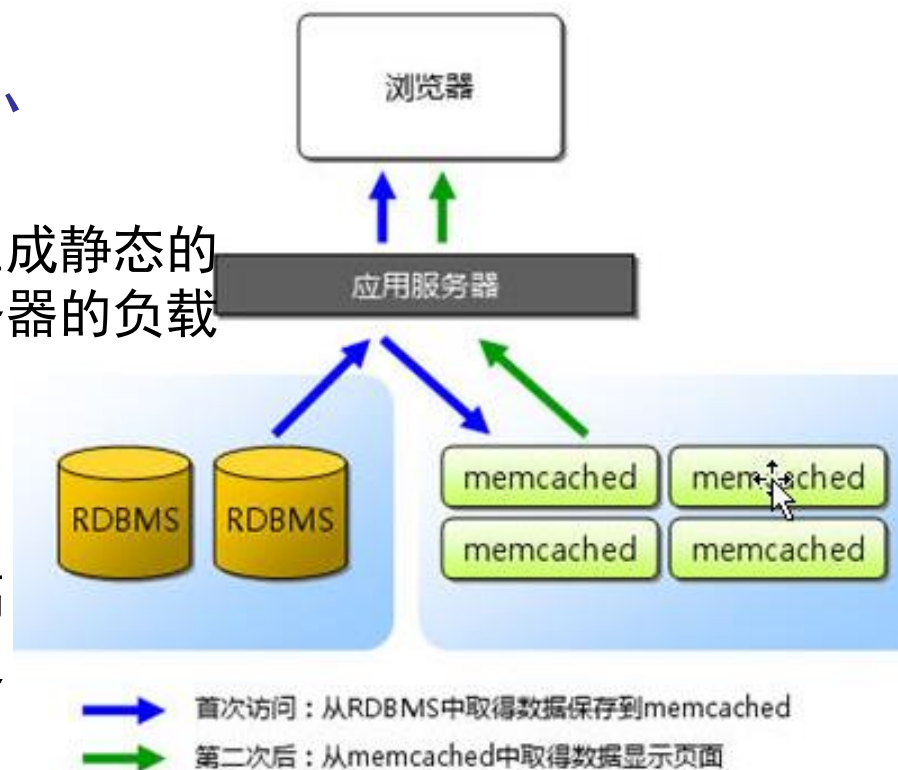
问题：如何确定旧的页面与新的内容相同？

- 1) 利用GET消息头的expiry判断是否过期，例如浏览器根据过去的页面更新间隔预测
- 2) 询问服务器缓存的副本是否过期



网页缓存技术

- 网页缓存技术分为客户端缓存、服务端缓存
- 客户端缓存分为：浏览器缓存、网关或代理服务器缓存
 - 如果启用浏览器缓存，则用户在访问同一个页面时，将不再从服务器下载，而是从本机缓存目录中读取页面然后展现；缺陷是读取旧的页面
- 服务端缓存分为：页面缓存、数据缓存、数据库缓存
 - 页面缓存：将动态页面直接生成静态的页面放在服务器端，节约服务器的负载
 - 数据缓存：缓存数据存放在服务器的内存或文件中
 - 数据库缓存：由数据库提供，可以对表建立高速缓存，提高对经常访问的数据的访问速度





流媒体应用与协议

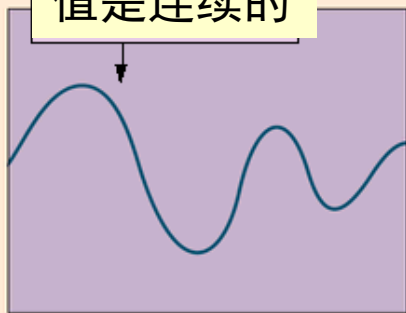
- 流媒体(streaming media): 音频/视频数据, 特点
 - “边下载边播放”
 - 下载结束后, 在用户的硬盘上不保存播放内容
- 数字音频与数字视频
- 三类服务
 - 存储的流媒体、实况的流媒体、交互式流媒体
- 传输需要哪些协议?
 - RTP、RTSP、RTCP、SIP
- 媒体播放器

数字音频：语音信号的数字表示

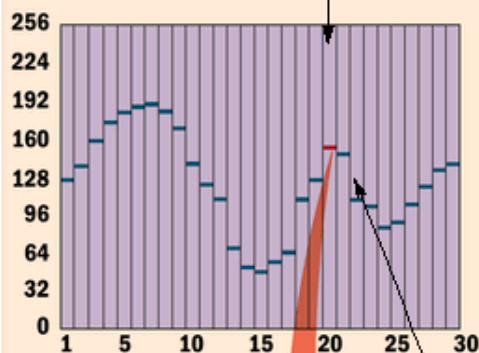
FIGURE 10-18

Sampling a

模拟声波为平滑曲线，值是连续的



波的数字化就是将波形纵向分割为多个，即采样。如每秒30个样点



Sample	Sample Height (Decimal)	Sample Height (Binary)
1	130	10000010
2	140	10001100
3	162	10100010
4	175	10101111
5	185	10111001
6	188	10111100
7
...
12	115	01110011
13	68	01000100
14	50	00110010
15	45	00101101
16	60	00111100
17	66	01000010
18	115	01110011
19	129	10000001
20	155	10011011
21	153	10011001
22	115	01110011

将样点的高度值转换为二进制并保存。如第20个样点的十进制155保存为10011011

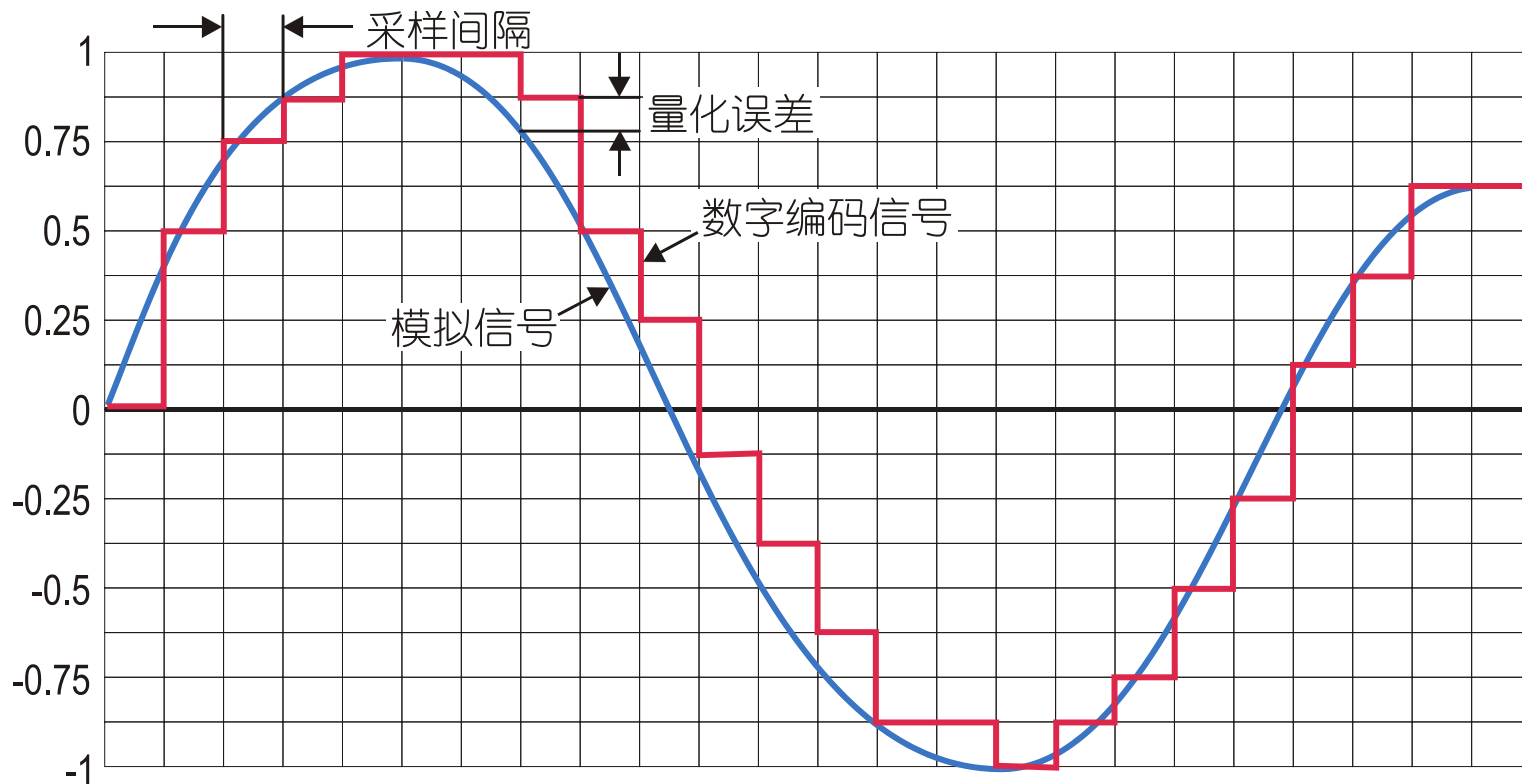
采样：对语音信号在时间轴上分段，将每秒采样的次数定义为**采样率**

量化：用一定的比特数表示样本幅度，将表示每个样值的比特数定义为**量化位数**

- 例如8KHz采样，8比特量化，数据率有64Kbps

数字音频

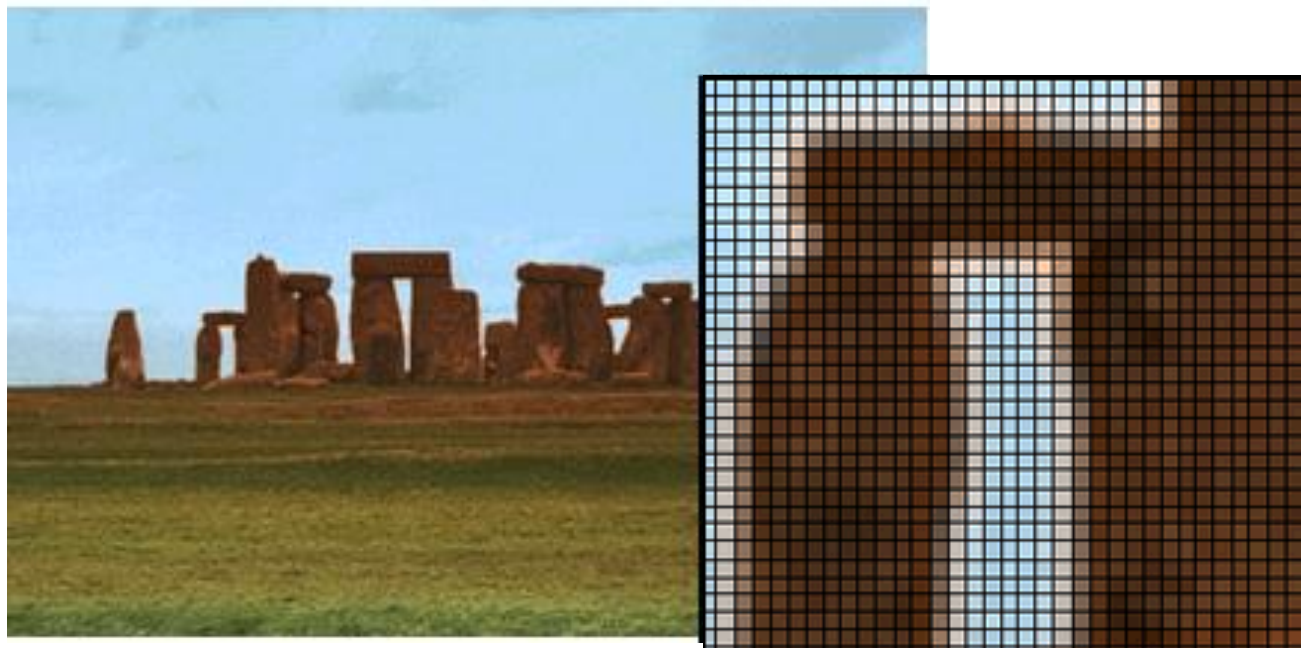
采样及量化会导致信号失真，损伤语音质量



奈奎斯特采样定理：当采样率为信号最高频率的2倍时，采样之后的数字信号完整地保留了原始信号中的信息

从数字图像到数字视频

- 将图像分割成像素（点阵），对像素的亮度/颜色（称为像素值）进行编码；保存每个像素值



- 数字视频：在时间轴上连续的数字图像——帧，例如24帧/s





数字音频/视频的特点

■ 数据量大

- 影响因素：时长，采样率、帧率、分辨率、量化位数等

■ 产生的数据与时间有关

- 数据传输要求实时性及传输数据率

■ 数据压缩技术：降低数据量及数据率

- 利用数据之间的相关性
- 利用人的听觉、视觉的不敏感性



典型的语音编码标准

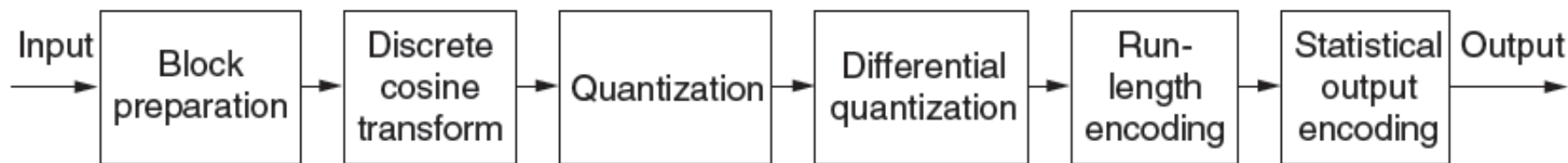
名称	速率/kbps	帧长 /ms	延时 /ms	复杂度 /ms	MOS 评分
G.711 (PCM)	64	-	0.125	-	4.5
G.723.1	5.3/6.4	30	37.5	16.5	3.8
GSM EFR	12.2	20	20	5	3.7
G.729	8	10	15	20	3.9
iLBC	13.3/15.2	30/20	>30/>20	20	4.0

(MOS = Mean Opinion Score)

数字视频压缩(1)

- 联合图像专家组JPEG标准，用于24位/像数的RGB数字图像压缩

JPEG有损编码过程



- 因人眼对亮度信号的敏感度高干颜色信号，故实施信号变换

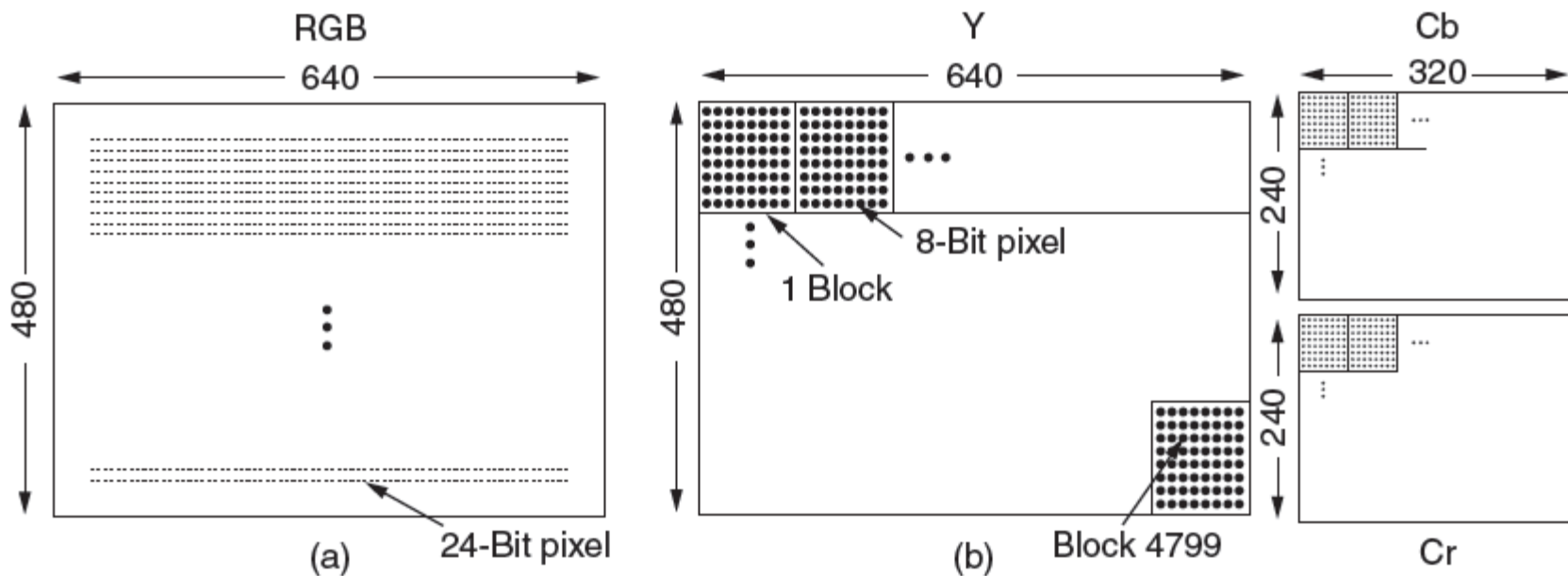
亮度分量 $Y = 16 + 0.26R + 0.50G + 0.09B$

色度分量 $Cb = 128 + 0.15R - 0.29G - 0.44B$

色度分量 $Cr = 128 + 0.44R - 0.37G + 0.07B$

数字视频压缩(2)

- 对Cb, Cr矩阵, 计算相邻4像素的均值, 将数据量降为1/4
- 对每 8×8 像素构成的一个块, 实施DCT (离散余弦变换)

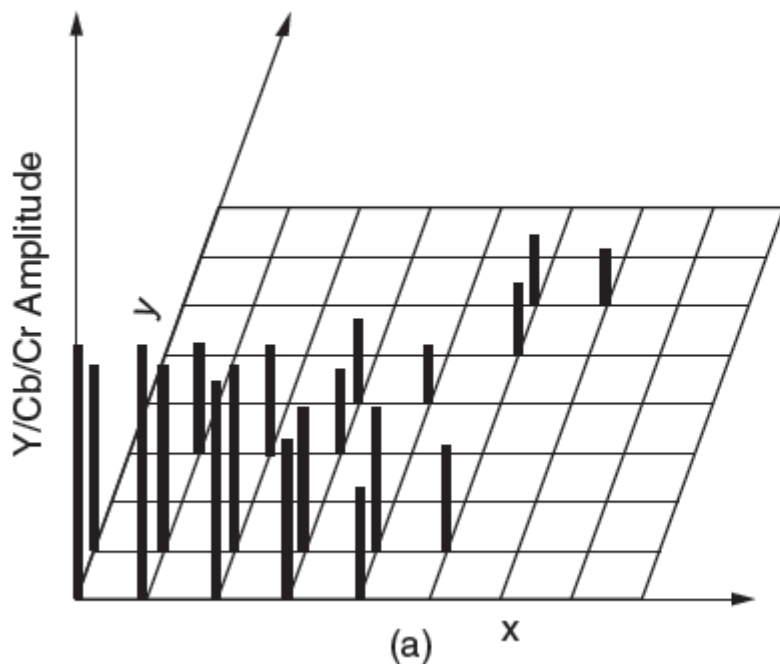


(a) RGB输入数据.

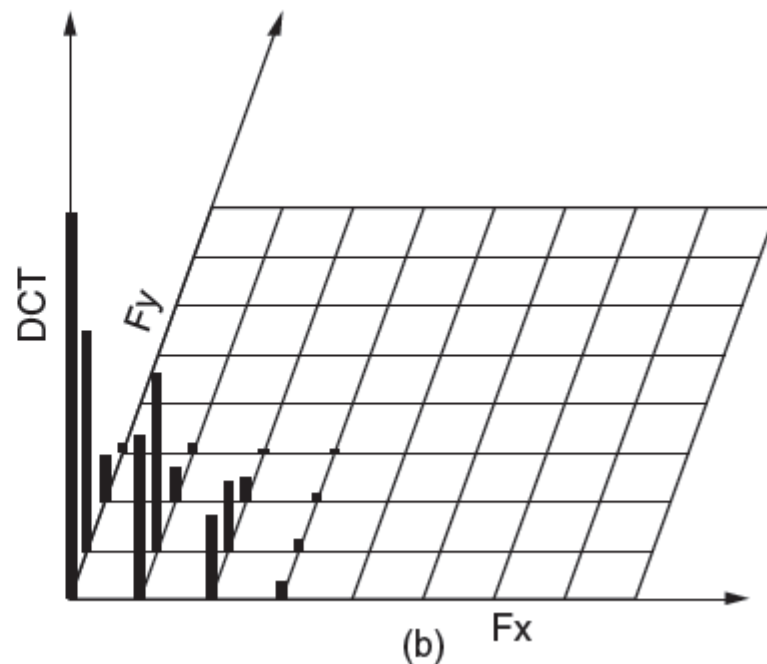
(b)块预处理之后

数字视频压缩(3)

元素 $(0, 0)$ 为每个块的均值，其他为空间频率的功率值，能量集中于低频部分



(a) Y矩阵的一块



(b) DCT系数

数字视频压缩(4)

DCT coefficients

150	80	40	14	4	2	1	0
92	75	36	10	6	1	0	0
52	38	26	8	7	4	0	0
12	8	6	4	2	1	0	0
4	3	2	0	0	0	0	0
2	2	1	1	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Quantization table

1	1	2	4	8	16	32	64
1	1	2	4	8	16	32	64
2	2	2	4	8	16	32	64
4	4	4	4	8	16	32	64
8	8	8	8	8	16	32	64
16	16	16	16	16	16	32	64
32	32	32	32	32	32	32	64
64	64	64	64	64	64	64	64

Quantized coefficients

150	80	20	4	1	0	0	0
92	75	18	3	1	0	0	0
26	19	13	2	1	0	0	0
3	2	2	1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

- 对DCT的系数进行量化：除以量化表对应的权值，缩小高频分量的值
- 将每块 (0, 0) 的与前一块的相减，减少其值的小大
- 按“Z”字形排列64个元素值，使得更多的0连续排列
- 对排列的数值采用霍夫曼码游程编码

150	80	20	4	1	0	0	0
92	75	18	3	1	0	0	0
26	19	13	2	1	0	0	0
3	2	2	1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

数字视频压缩：MPEG压缩

- MPEG，同时压缩音频和视频
- 视频压缩，利用图像的空间及时间相关性
 - 利用空间相关性：详见JPEG
 - 利用时间相关性：对相邻帧的图像求差值，再采用JPEG
 - 输出三类帧
 - I帧：帧内编码
 - P帧：预测帧与前一帧的逐块差值
 - B帧：双向帧与前一帧和后遗症的逐块差值

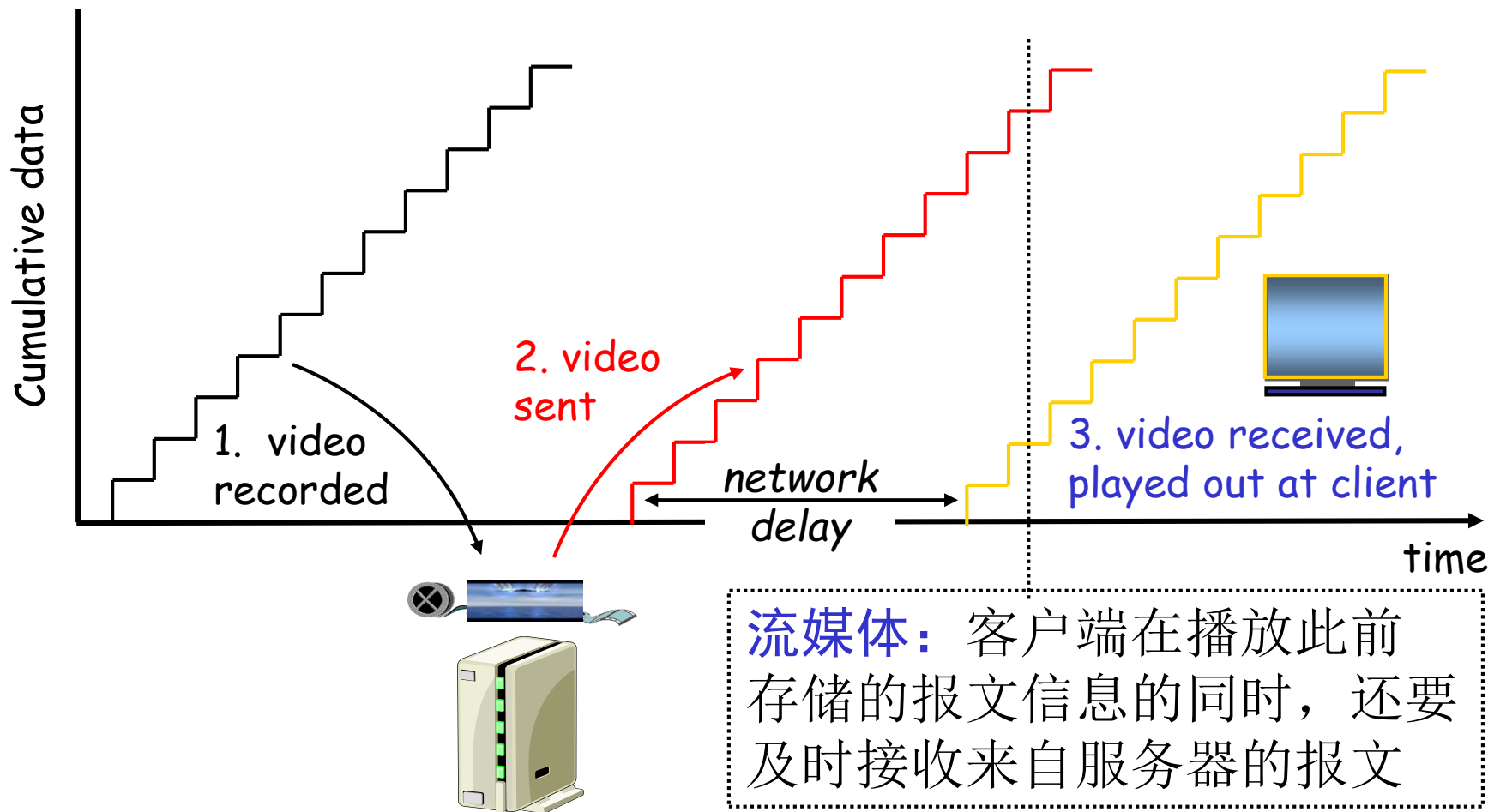




流媒体应用与协议

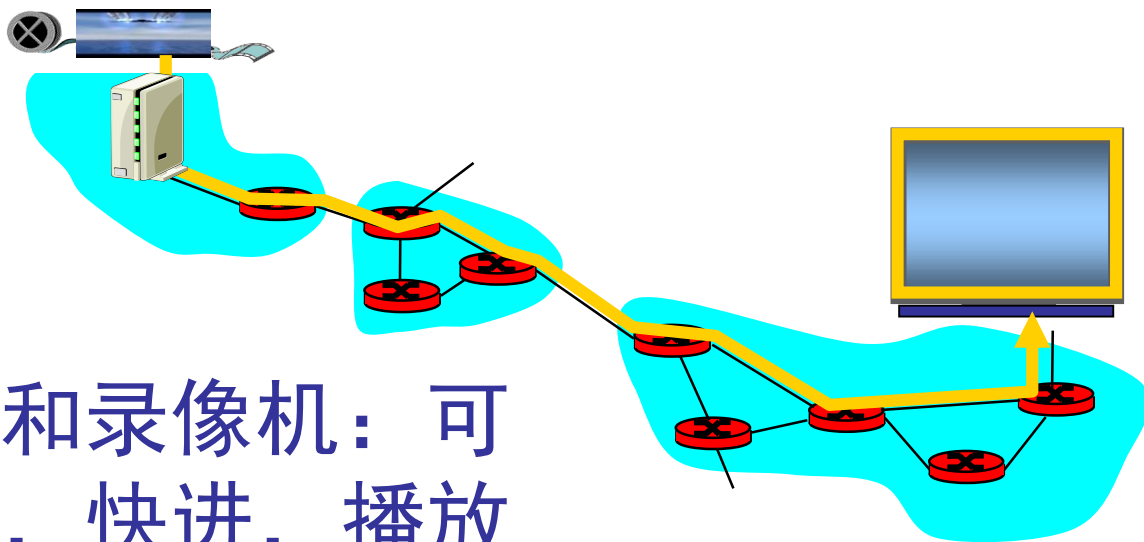
- 流媒体(streaming media): 音频/视频数据, 特点
 - “边下载边播放”
 - 下载结束后, 在用户的硬盘上不保存播放数据
- 数字音频与数字视频
- 三类服务
 - 存储的流媒体、流式直播媒体、实时会议
- 传输需要哪些协议?
 - RTP、RTSP、RTCP、SIP
- 媒体播放器

流媒体



流媒体：客户端在播放此前存储的报文信息的同时，还要及时接收来自服务器的报文

存储的流媒体：弱交互



- 类似于录音机和录像机：可以暂停，回放，快进，播放
 - 延迟为10s
 - 命令响应时间为1~2s
- 时间限制：根据初始记录的时序实时播放

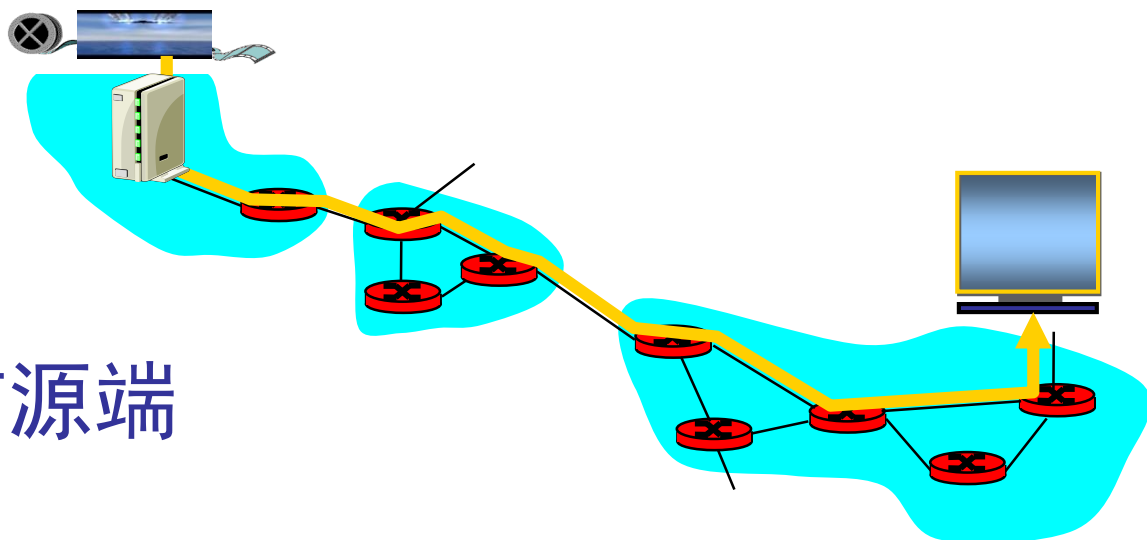
流式直播媒体：连续播放

存储的媒体：

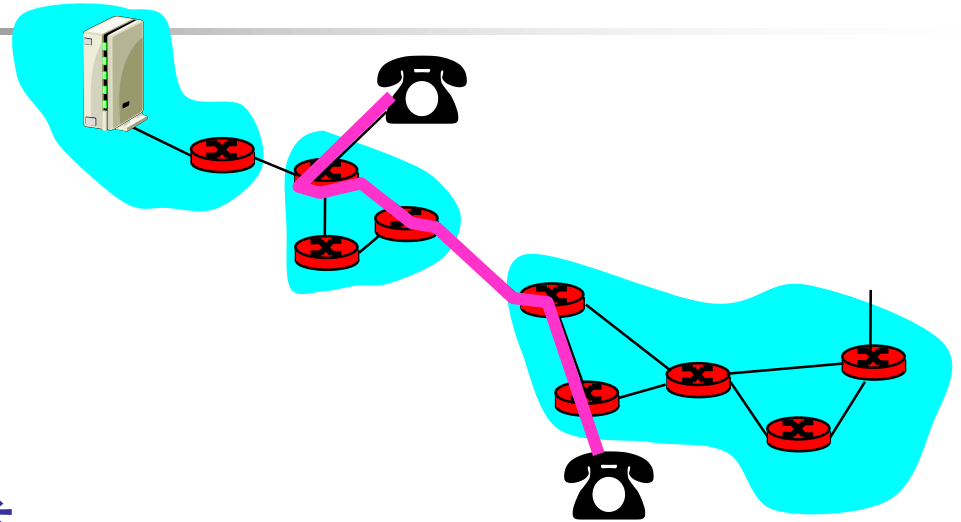
- 媒体存储在信源端
- 传递给客户

流：

- 当客户接收到数据后，根据初始记录的时序连续播放，类似于传统的电台及电视台

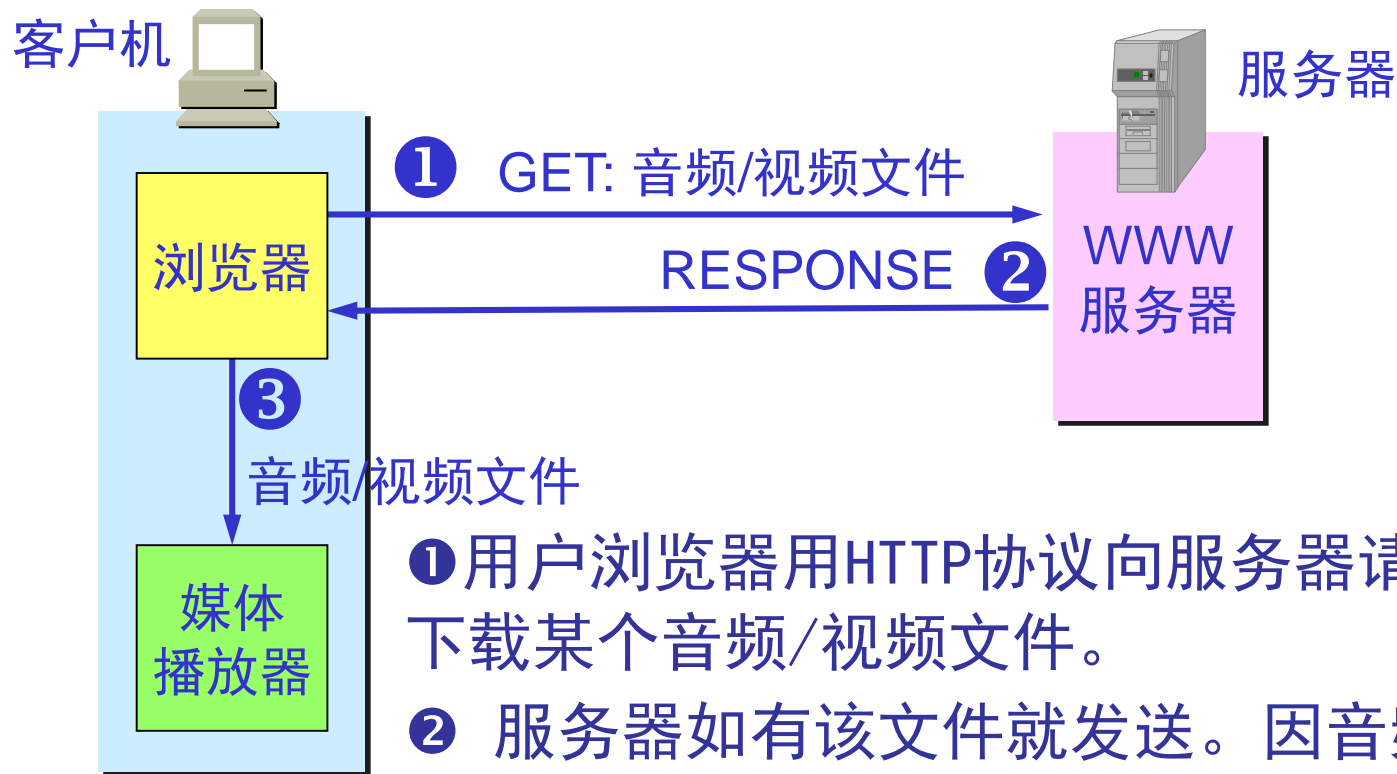


实时会议：强交互



- 应用：IP 电话，视频会议
- 端到端的延迟要求：
 - 语音：<150ms为优；<400 ms可接受
 - 包括应用层封装分组（信源编码）及网络层传输延迟（发送、传播、处理、排队）
 - 更高的延迟将影响交互性

流式存储的音频/视频



① 用户浏览器用HTTP协议向服务器请求下载某个音频/视频文件。

② 服务器如有该文件就发送。因音频/视频文件很大，下载时间可能很长。

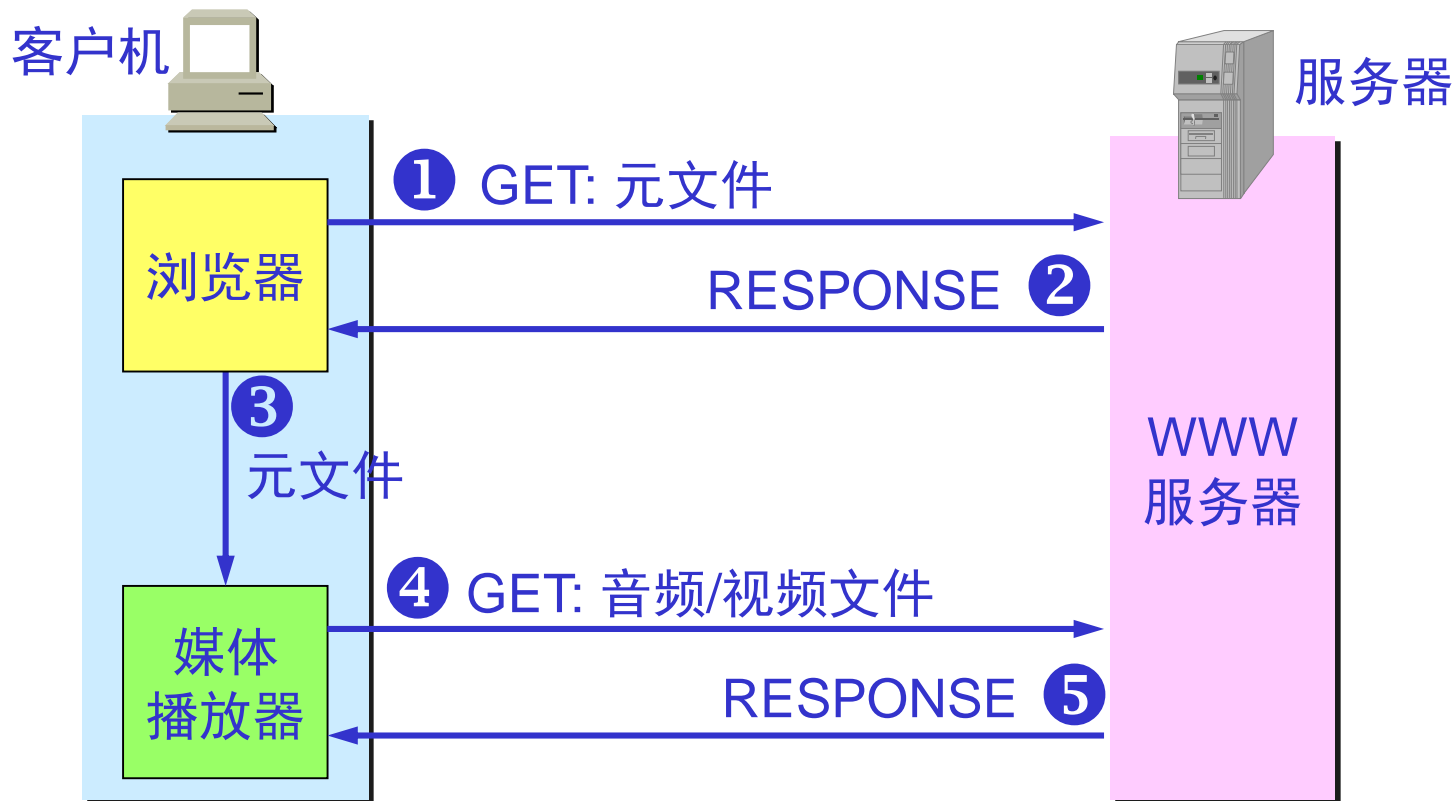
③ 当浏览器完全收下文件后，才传给本机的媒体播放器解压缩并播放。

具有元文件的Web服务器

元文件非常小，含有音频/视频文件的URL

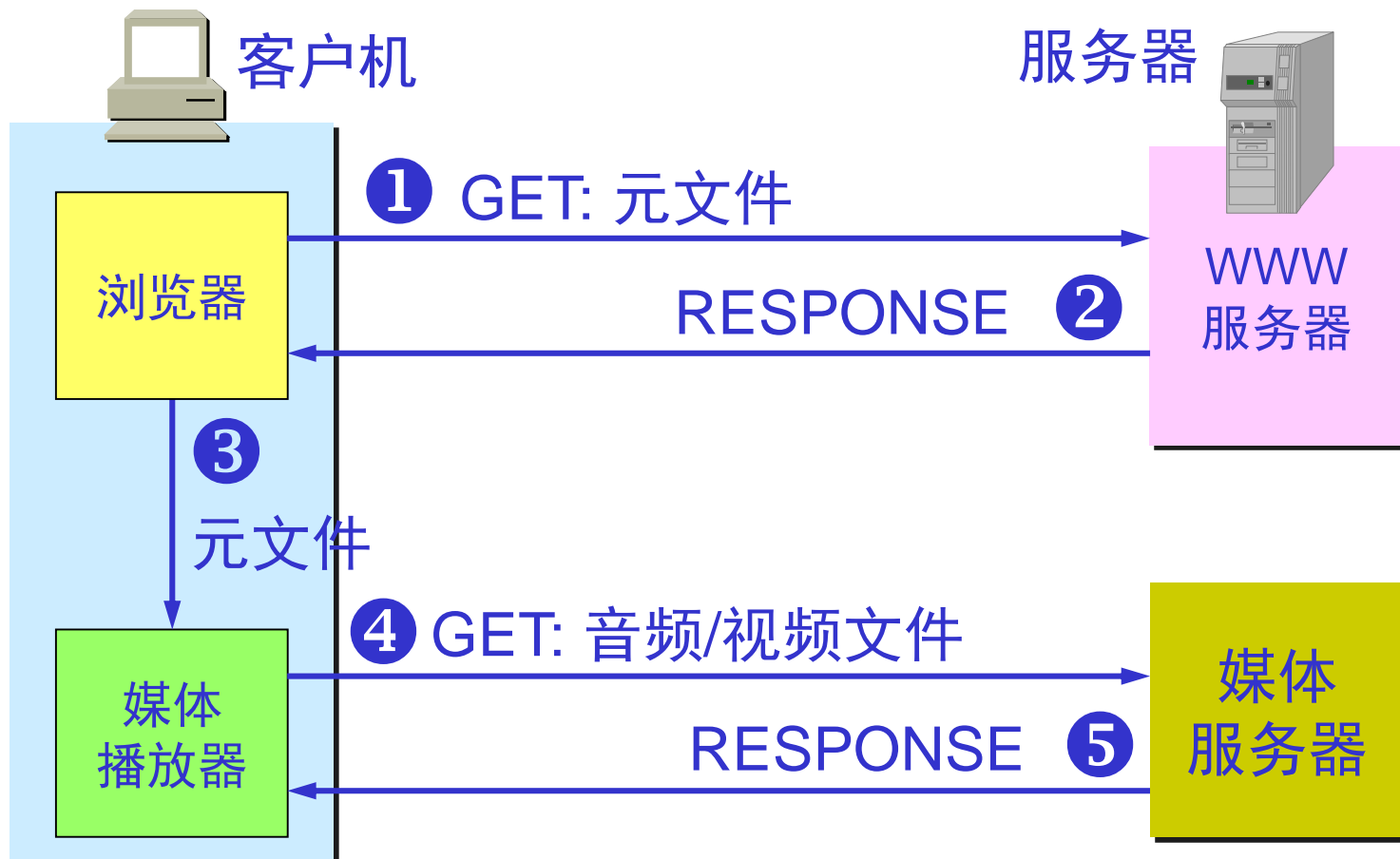
④ 媒体播放器用元文件向Web服务器发送HTTP请求

⑤ Web服务器发送HTTP响应，把该音频/视频文件发送给媒体播放器。媒体播放器边下载边解压缩边播放。



使用媒体服务器

- **媒体服务器**也称为**流式服务器**(streaming server)，它支持流式音频和视频的传送。
- 媒体服务器和媒体播放器之间采用RTSP协议进行交互

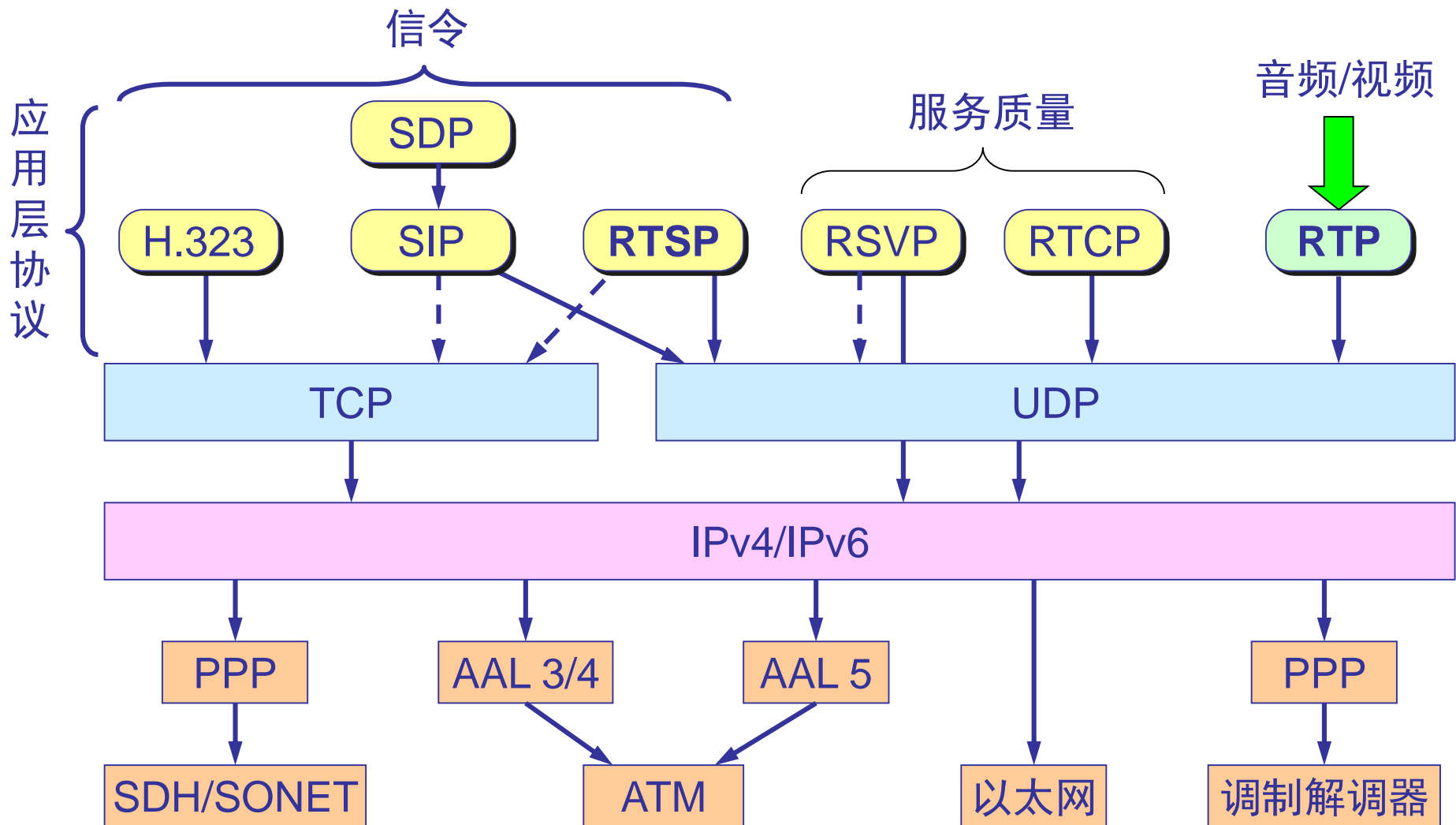




流媒体应用与协议

- 流媒体(streaming media): 音频/视频数据, 特点
 - “边下载边播放”
 - 下载结束后, 在用户硬盘上不保存播放的内容
- 数字音频与数字视频
- 三类服务
 - 存储的流媒体、直播的流媒体、交互式流媒体
- 传输需要哪些协议?
 - RTP、RTSP、RTCP、SIP
- 媒体播放器

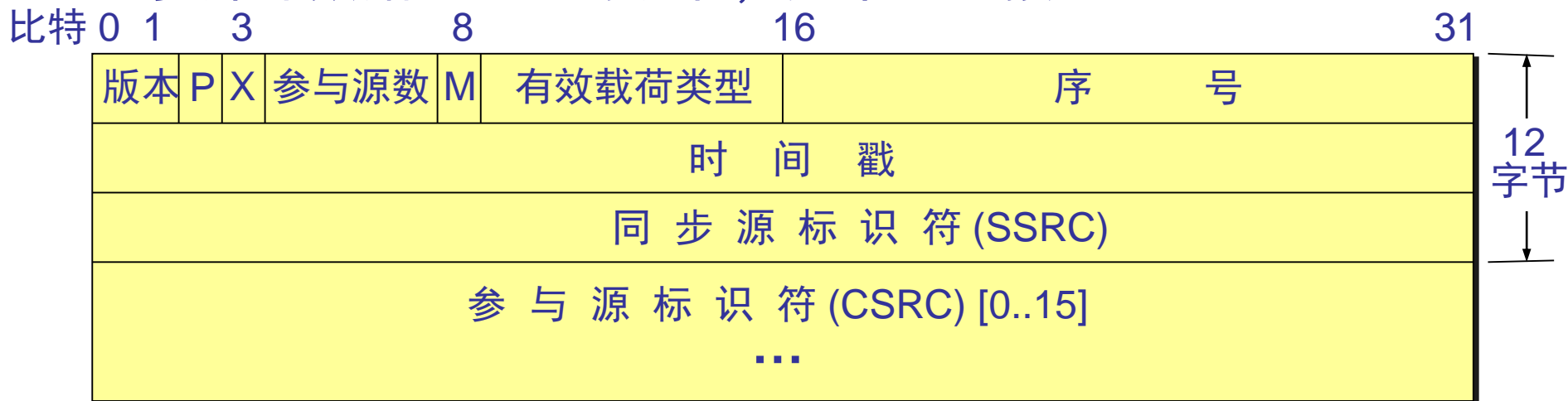
互联网多媒体协议



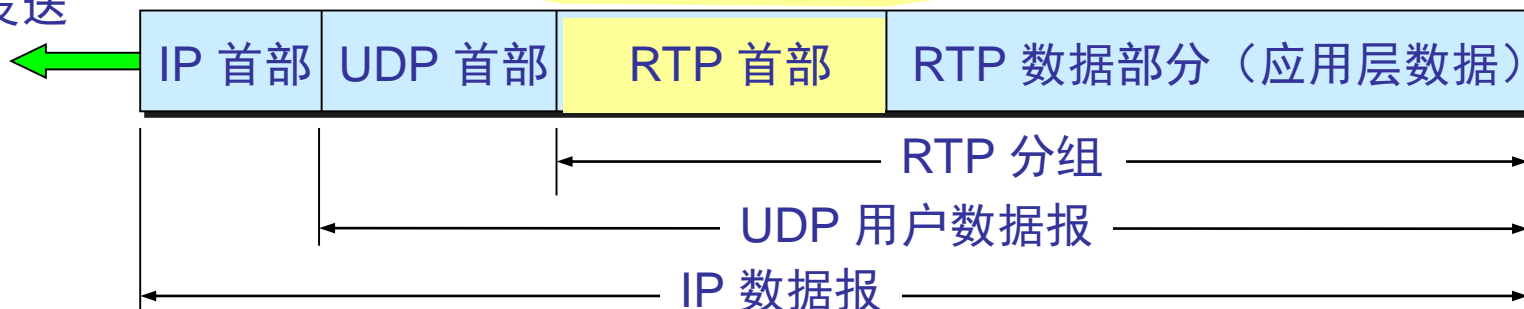
实时传输协议 RTP

(Real-time Transport Protocol)

- RTP 为实时应用提供端到端的传输服务
- 多媒体数据经RTP封装后，交给UDP接口（Socket）



发送





RTP 首部

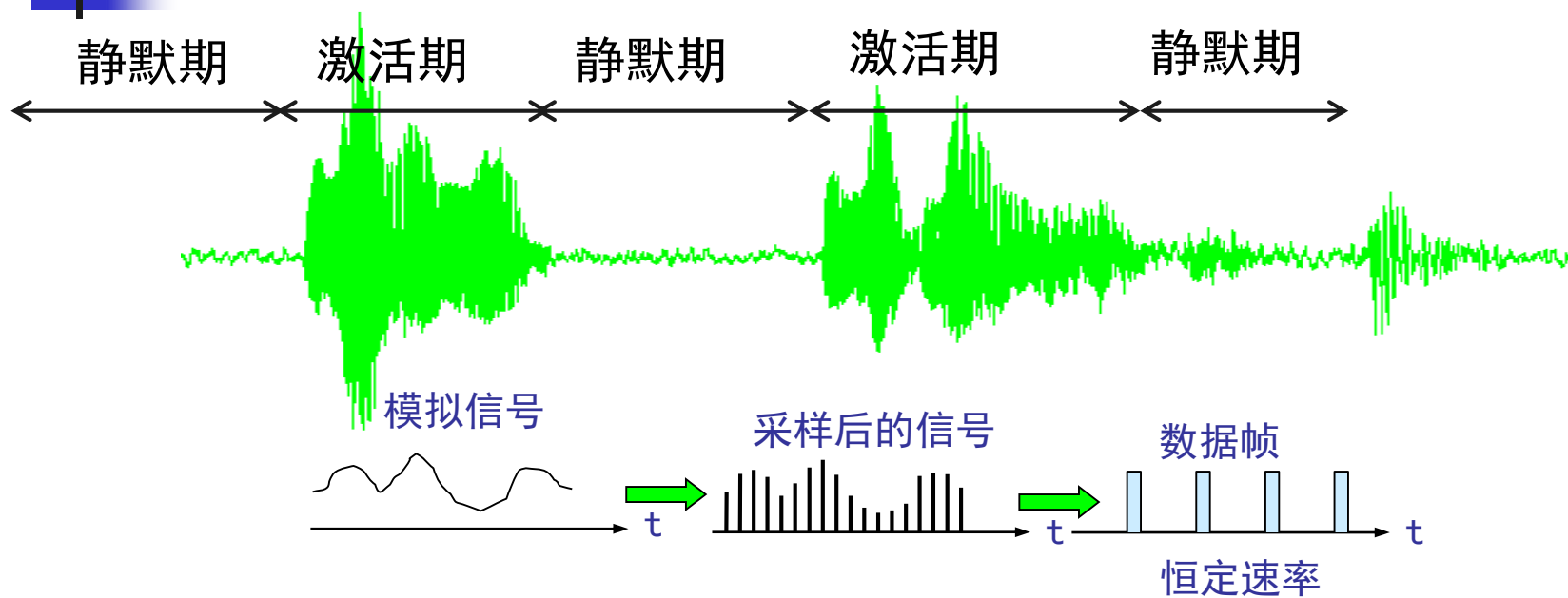
- 有效载荷类型：指示信源编码类型
 - 0: PCM , 64 kbps
 - 3: GSM, 13 kbps
 - 7: LPC, 2.4 kbps
 - 26: Motion JPEG
 - 31: H.261
 - 33: MPEG2 video
- 序号：每发送一个RTP报文，序号加1；接收端用序号检测丢包并恢复



RTP 首部(2)

- 时间戳：在RTP数据中第1个字节的采样时刻
 - 对于音频，时间戳时钟增加值为一个采样间隔（例如，对于8KHz采样时钟，增加值为 $125\ \mu\text{s}$ ）
 - 如果话音数据分组有160个采样值；当处于激活期时，则每个RTP分组的时间戳增加采样间隔 $\times 160$ 倍；当处于静默期时，时间戳也以恒定速率增加。
- 同步源标识 (SSRC)：标识RTP流的信源，在RTP会话中的每个流有唯一的SSRC

数字音频



- 话音音频：激活期，静默期
 - 激活期，帧间隔时间固定，如20ms
 - 静默期，无信息，可以不传输数据
- 语音帧+RTP头（序号、类型、时间戳）——数据块
- 数据块+UDP头——UDP报文

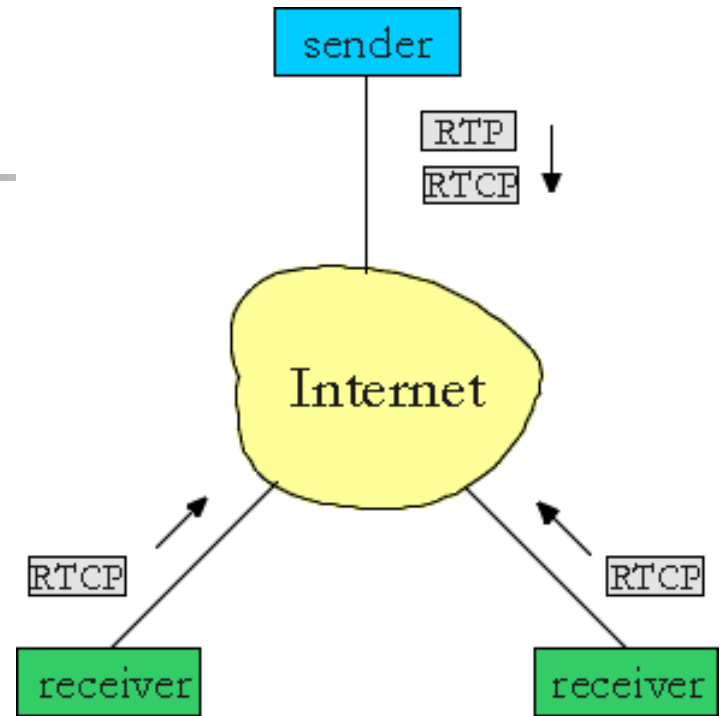


RTCP——RTP控制协议

(RTP Control Protocol)

- RTCP 功能：服务质量监测与反馈、媒体流之间的同步，多播组的成员标识，与RTP配合使用。
- RTP会话参加者周期地传送RTCP，携带服务质量信息，如发送分组数、丢包率、延迟的变化
- RTCP 五类报文
 - 结束报文BYE：关闭一个数据流
 - 特定应用报文APP：定义新的分组类型
 - 接收端报告RR：接收端用多播方式周期地发送
 - 发送端报告SR：发端用多播方式周期地发送
 - 信源描述报文SDES：描述会话参加者

RTCP (续)



- 一个 RTP会话： 一般为一个多播组；所有属于该会话的 RTP/RTCP使用组播地址
- 通过端口号区分不同会话的RTP， RTCP
- 限制RTCP数据量， 每个参加者根据会话参加者的数量增减RTCP数据量



RTCP 分组

接收端报告RR：接收端用多播方式周期地发送

- 分组丢失比例，最新序号，平均延迟的变化

发送端报告SR：发端用多播方式周期地发送

- RTP流的SSRC，当前时间，发送分组数，发送字节数

信源描述SDES：描述会话参加者

- 发送端e-mail地址、名字，与RTP流有关的SSRC
- 建立SSRC到用户/主机名的映射



流的同步

RTCP在一个RTP会话内同步不同的媒体流

- 例如视频会议，每个发端产生两个RTP流：一个视频流，一个音频流
- 通过RTP时间戳将视频与音频的采样时刻绑定
- 在最新RTCP发端报告SR中，含有
 - RTP分组时间戳
 - 分组产生时间
- 收端利用这一关联信息，在播放音频及视频时，保证两个流的同步



RTCP 带宽调整

RTCP 试图限制其发送的数据量，只占会话带宽的5%

- 若一个发端以2 Mbps发送视频，则 RTCP 试图限制其速率到100 Kbps.
- RTCP分配带宽的75%给接收端，25%给发端
- 发端的RTCP速率为25Kbps
- 所有接收端之间共享75Kbps：若R个接收端，则每个接收端发送RTCP的速率 $(75/R)$ Kbps
- 参加者确定RTCP分组发送间隔：
= RTCP平均分组长度/分配速率



实时流媒体协议RTSP

(Real-Time Streaming Protocol)

- RTSP 是多媒体播放控制协议，是**媒体播放器**与**媒体服务器**之间的控制协议，使用户可以控制下载数据的播放，如暂停/继续、后退、前进等。
- RTSP 的控制功能在专用软件中实现，如**媒体播放器**、**媒体服务器**

RTSP: RFC 2326

- 控制消息的传输采用UDP或TCP
- 使用端口554，与媒体流的传输端口不同



RTSP的控制功能

从播放器到服务器的RTSP命令

Command	Server action
DESCRIBE	List media parameters
SETUP	Establish a logical channel between the player and the server
PLAY	Start sending data to the client
RECORD	Start accepting data from the client
PAUSE	Temporarily stop sending data
TEARDOWN	Release the logical channel

客户机



服务器



浏览器

万维网
服务器

媒体
播放器

媒体
服务器

① GET: 元文件

RESPONSE ②

③
元文件

④ SETUP

RESPONSE ⑤

⑥ PLAY

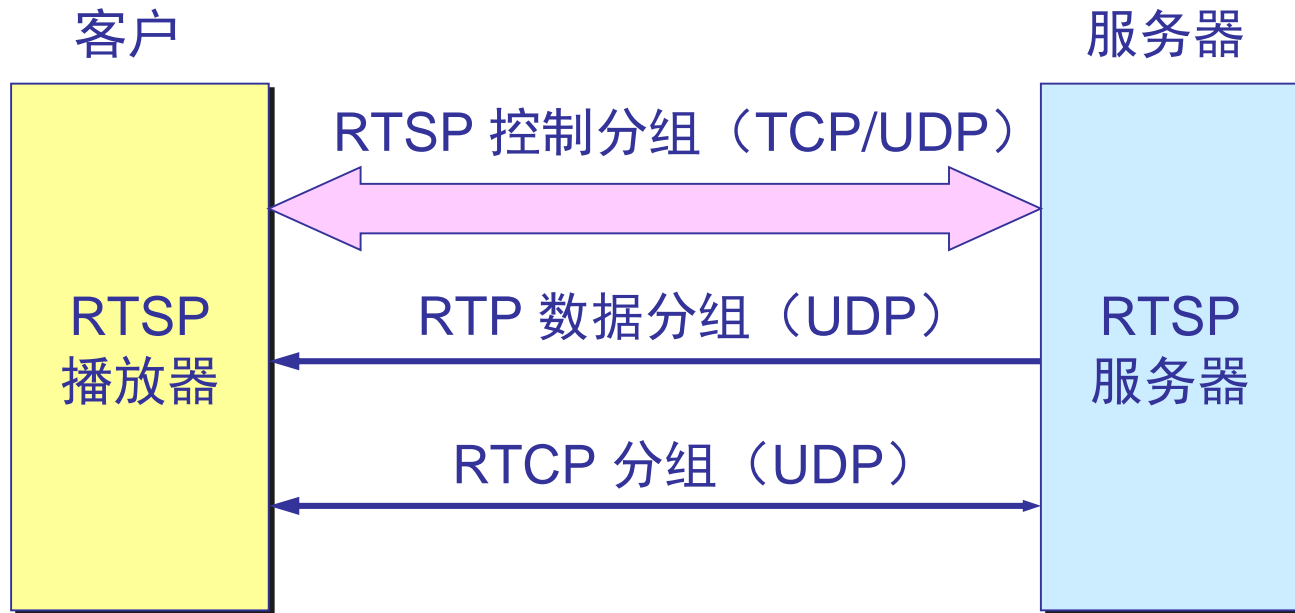
RESPONSE ⑦

⑧ TEARDOWN

RESPONSE ⑨

音频/视频流

RTSP与RTP及RTCP的关系



RTP：用于传送多媒体流；

RTSP：播放器控制媒体服务器流的传送；

RTCP：用于监测服务质量、同步媒体流、标识多播组成员等



SIP 协议

(Session Initiation Protocol)

- 互联网可以支持电话呼叫，视频会议
- SIP用于IP电话的信令和服务质量，与RTP和RTCP协议配合
- SIP系统有两种构件：用户代理和网络服务器
- 用户代理：发起呼叫或接受呼叫
- 网络服务器：分为代理服务器和重定向服务器
 - 代理服务器转发用户呼叫请求，最后到达被叫用户
 - 重定向服务器通知下一跳代理服务器的地址



SIP:[RFC 3261]

- 用名字或e-mail地址标识用户，而不用电话号码，无论被叫用户漫游到哪里，都可以找到

SIP提供的功能

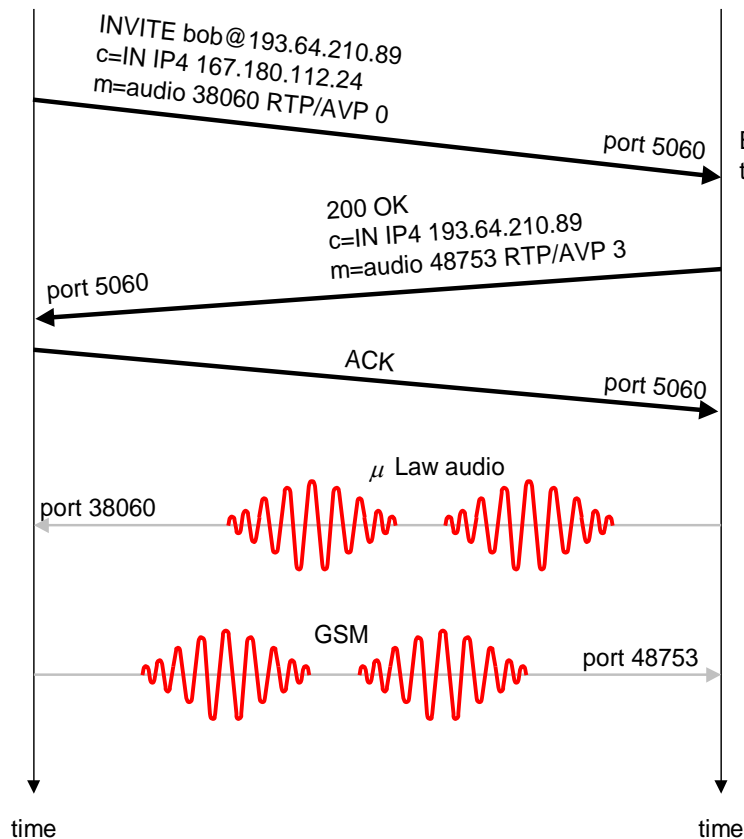
- 建立呼叫
 - 使被叫知道是谁在呼叫他
 - 主叫、被叫可以协商媒体类型、编码方式
 - 结束呼叫
- 确定被叫的IP地址
 - 将用户标识映射为IP地址
- 呼叫管理
 - 在呼叫期间增加新媒体流、改变编码方式
 - 邀请其他人加入、呼叫转移以及保持通话

当已知IP地址时，建立一次呼叫



167.180.112.24

193.64.210.89



- Alice的SIP邀请：指示其端口号，IP地址，接收编码类型(PCM μ 律)
- Bob的200 OK：指示其端口号，IP地址，期望的编码类型(GSM)
- SIP消息可采用TCP或UDP发送；在本例中采用RTP/UDP
- 缺省SIP端口号为5060
- 编码协商：假设Bob不支持PCM μ 律编码，则将应答606表示不接受请求，而接收者Alice可发送新的邀请消息，声明其他的编码
- 拒绝呼叫：Bob可以用“忙”，“不在”，“欠费”，“禁用”等拒绝呼叫，也可能因媒体采用RTP或其他协议等原因拒绝呼叫



名字翻译及用户定位

当主叫要呼叫被叫，但只有被叫名字或邮件地址时

- 需要得到被叫**当前**主机的IP地址：
 - 用户改变位置
 - DHCP 协议
 - 用户有不同的IP设备（PC、PDA、车载通信设备）
- 产生的原因
 - 被叫在单位或在家
 - 主叫不愿意拨打老板的家庭电话
 - 被叫的状态（主叫在呼叫被叫，而被叫正与第三人通话）

由SIP服务器提供服务

- SIP 注册服务器
- SIP 代理服务器



SIP 注册

- 当 Bob 启动 SIP 客户程序时，由客户程序发送 SIP REGISTER 消息给 Bob 的注册服务器

Register Message:

```
REGISTER sip:domain.com SIP/2.0  
Via: SIP/2.0/UDP 193.64.210.89  
From: sip:bob@domain.com  
To: sip:bob@domain.com  
Expires: 3600
```



SIP Proxy

- Alice向其代理服务器发送邀请消息
 - 含有地址 sip:bob@domain.com
- 代理服务器负责将SIP消息路由到被叫
 - 有可能经过多个代理服务器
- 被叫发送响应经过同样的多个代理服务器
- 代理服务器返回SIP响应消息给Alice
 - 含有Bob的IP地址
- 代理服务器就像本地的DNS服务器

举例

主叫jim@umass.edu呼

叫keith@upenn.edu

(1) Jim发送INVITE
消息给umass的SIP代理

(2) 代理转发消息到
upenn的注册服务器

(3) upenn服务器返回重
定向信息, 指示其与
keith@eurecom.fr联系

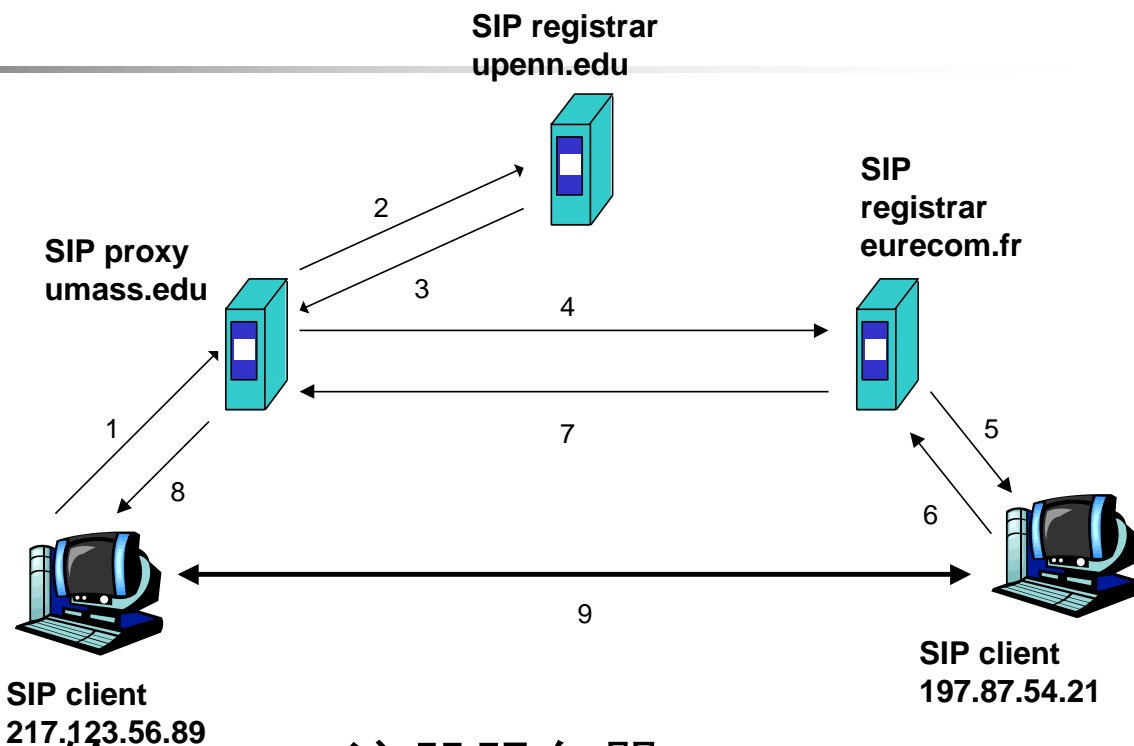
(4) umass代理发送INVITE给eurecom注册服务器

(5) eurecom注册服务器转发INVITE消息到keith的SIP客
户程序197.87.54.21; (6-8) 返回SIP响应

(9) 直接在客户之间交换多媒体信息

注释: SIP也有ack信息, 但此处没有给出

SIP也可以支持移动性, 通过注册服务器、重定向





流媒体应用与协议

- 流媒体(streaming media): 音频/视频数据, 特点
 - “边下载边播放”
 - 下载结束后, 在用户的硬盘上不保存播放的内容
- 数字音频与数字视频
- 三类服务
 - 存储的流媒体、直播的流媒体、交互式流媒体
- 传输需要哪些协议?
 - RTP、RTSP、RTCP、SIP
- 媒体播放器

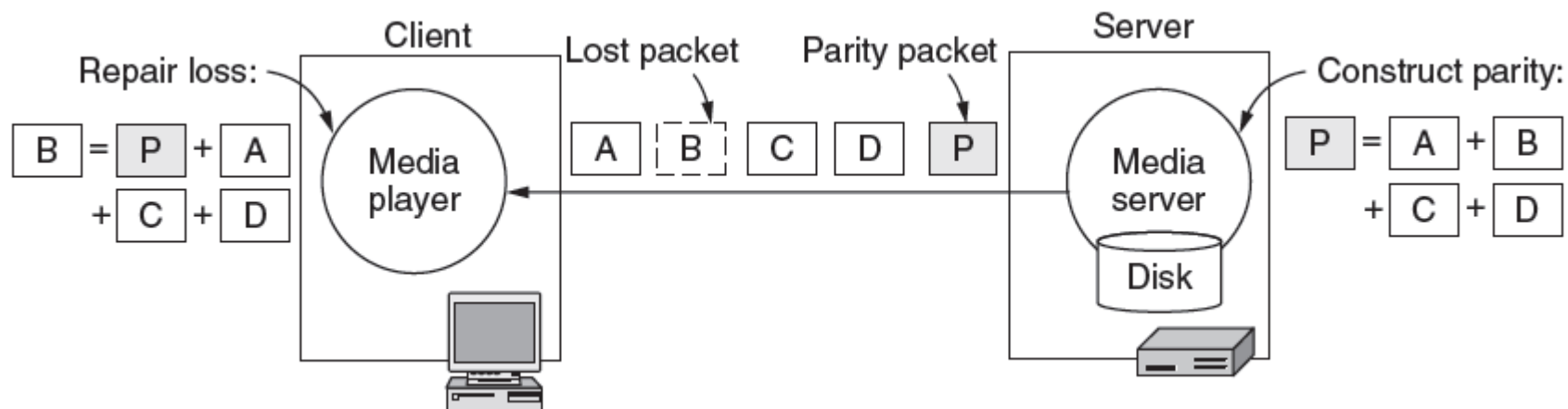


媒体播放器

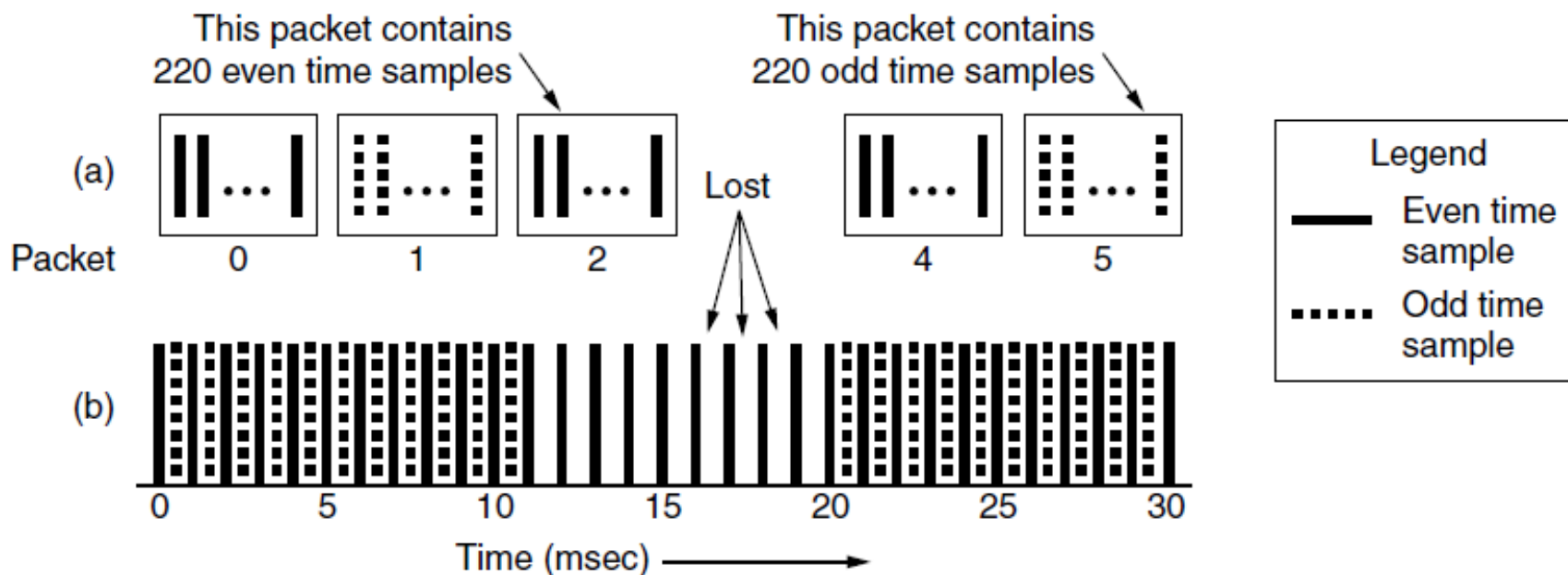
- 专用的软件，如media player
- 提供用户图形界面
- 提供交互功能——支持RTSP
- 解压缩
- 消除错误
- 缓存数据，消除抖动

媒体播放器——消除错误

- 使用FEC简单的纠错编码，如在服务器端每发送4个报文，构造一个检验报文（位的异或），与原有的4个报文一起发送
- 传输中，丢失报文B
- 接收时，利用异或重构报文B

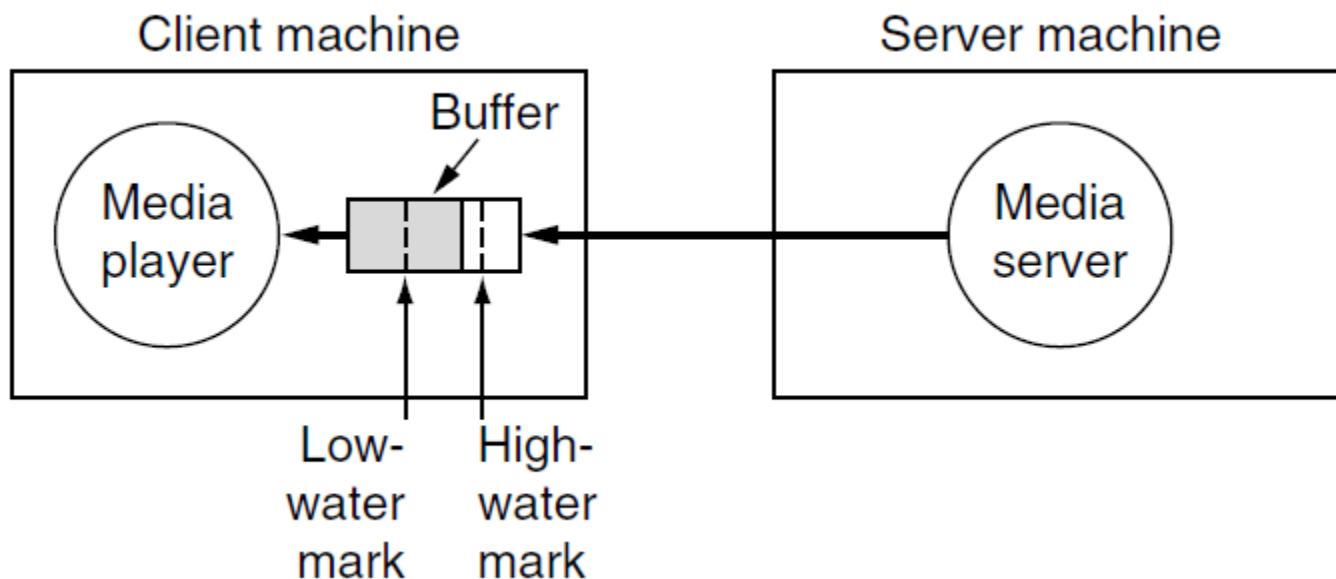


媒体播放器——消除错误



当数据携带奇偶交错的采样值时，丢失一个分组只是临时降低了分辨率，而不会造成播放时间的空隙

媒体播放器——缓存数据、消除抖动



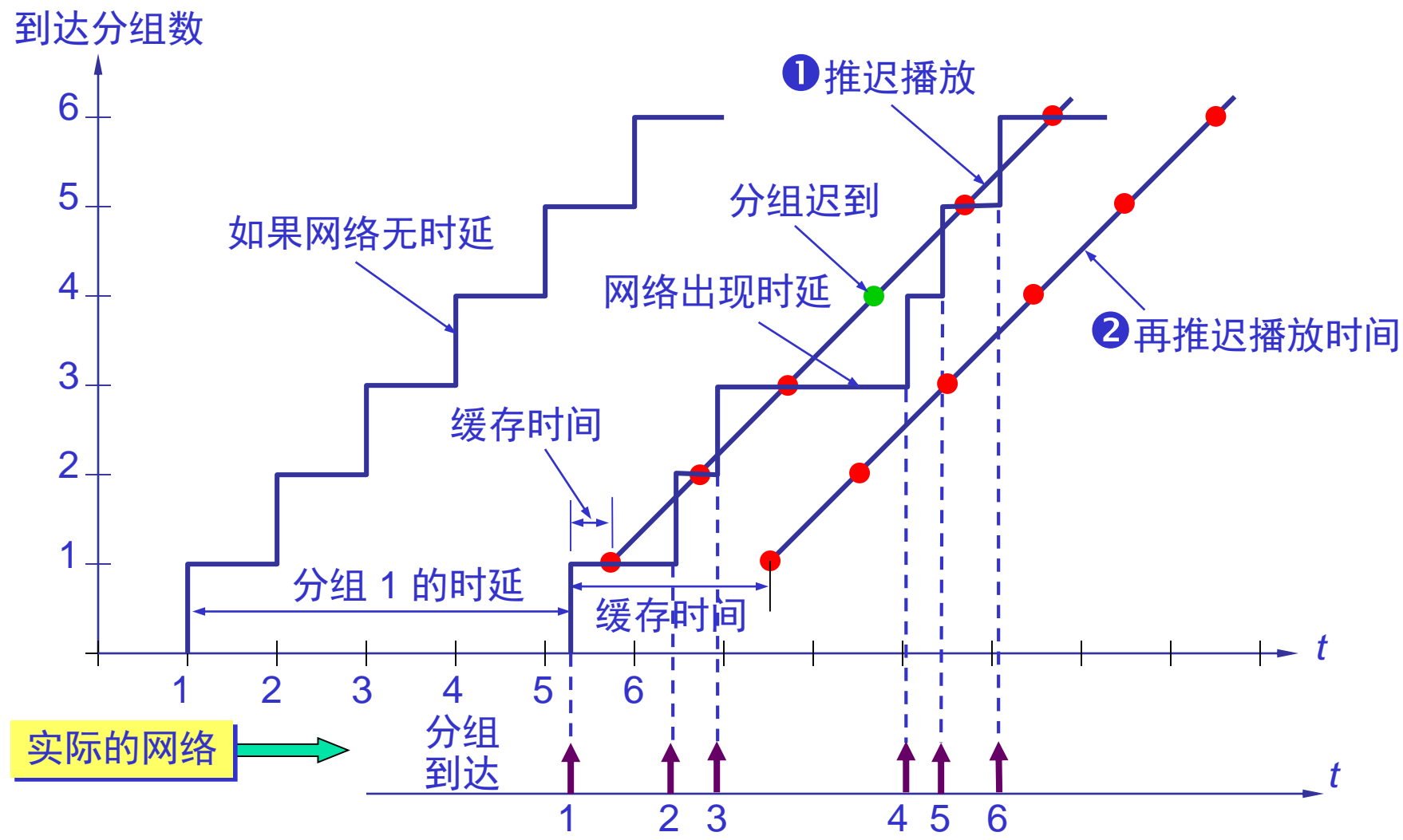
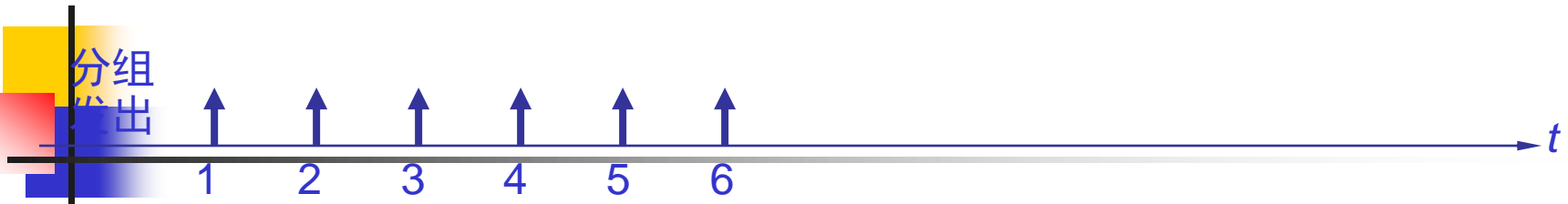
播放器缓存来自服务器的输入，播放缓存的内容，而不是直接播放接收的来自网络的数据

- 在接收端设置缓存；当缓存分组达到一定数量后再以恒定速率按序播放
- 缓存增加了**迟延**，但消除了**时延抖动**



网络电话：分组丢失及延迟

- **网络丢失：** IP报文丢失，因网络拥塞（路由器缓存溢出）
- **延迟丢失：** IP报文到达接收端播放器的时间太迟了
 - 延迟：数据处理时间，分组在网络中排队；（发端，收端）系统的延迟
 - 一般可容忍的最大延迟：400ms
- **可容忍延迟：** 与话音编码器有关，是可消除的丢包；可容忍的丢包率在1%~10%之间



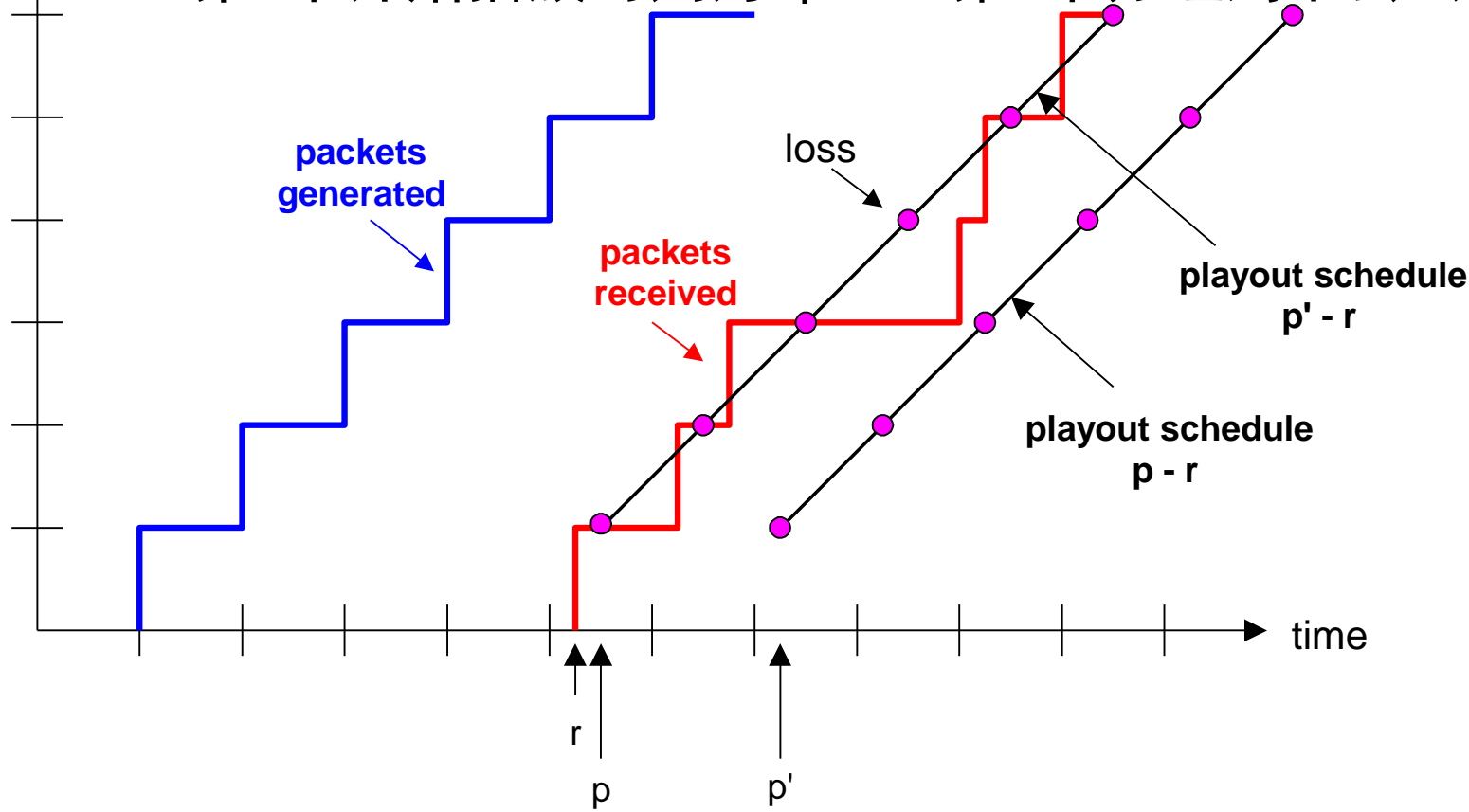


网络电话：固定的播放延迟

- 如果接收端收到话音块 q ms之后，再播放
 - 话音块有一时间戳：在 $t+q$ 时刻播放话音块
 - 话音块在 $t+q$ 之后达到：数据到达播放器太晚了，导致数据丢失
- 如何选择 q :
 - 更大的 q ：更少的分组丢失
 - 较小的 q ：更好的交互体验

固定的播放延迟

- packets
- 在激活期，发端每20 ms产生一个分组
 - 在时刻 r ，接收到第1个分组
 - 第1个开始播放时刻为 p ：第4个分组丢失
 - 第2个开始播放时刻为 p' ：第4个分组则不会丢失





自适应的播放延迟 (1)

- 目标：减少播放延迟，降低丢包率
- 方法：自适应调整播放延迟
 - 估计网络延迟，在语音激活期开始时，调整播放延迟
 - 压缩或加长静默期
 - 保证在激活期期间，播放间隔为 20 ms

t_i = timestamp of the i th packet

r_i = the time packet i is received by receiver

$r_i - t_i$ = network delay for i th packet

d_i = estimate of average network delay after receiving i th packet

在接收端，动态估计网络平均延迟：

$$d_i = (1 - u)d_{i-1} + u(r_i - t_i) \quad \text{其中, } u \text{ 为常量 (如 } u=0.01)$$



自适应的播放延迟(2)

- 平均时延的绝对偏差, v_i :

$$v_i = (1 - u)v_{i-1} + u |r_i - t_i - d_i|$$

- 对每个接收的分组, 估计 d_i , 计算 v_i
- 每个激活期开始的第1个分组, 播放时刻为:

$$p_i = t_i + d_i + Kv_i$$

其中, K 为一常量

- 在激活期期间的后续分组, 以固定间隔播放



自适应的播放延迟(3)

问题：接收端如何知道激活期开始的首个分组？

- 若没有丢失，接收端检查分组的时间戳（RTP 中）
 - 连续分组时间戳的差值 $> 20 \text{ ms}$ \rightarrow 激活期开始
- 可能出现分组丢失，接收端检查连续分组的时间戳以及序号
 - 连续的时间戳的差值 $> 20 \text{ ms}$ 并且序号连续， \rightarrow 激活期开始



小结

- 流媒体(streaming media): 音频/视频数据
- 数字音频与数字视频
 - 有实时性要求, 数据量大, 采用数据压缩技术
- 三类服务
 - 存储的流媒体、直播的流媒体、交互式流媒体
- 传输协议
 - RTP、RTSP、RTCP、SIP
- 媒体播放器
 - 用户界面、解压缩、消除差错、缓存及播放



练习题

- Web服务（通过阅读教材的第7章）
 - 网页、URL
 - 访问Web服务的过程：
 - 浏览器、服务器、HTTP
 - Web服务器如何处理多个请求？
 - Cookies的作用：利与弊
- 互联网上与流媒体服务相关的协议有RTSP, RTP, RTCP。试问这些协议的功能是什么？
- 媒体播放器的功能，它如何改善用户体验？