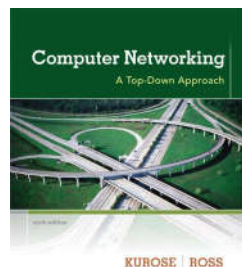


2. 多媒体网络

中国科学技术大学
自动化系 郑烜
改编自 Jim kurose, Keith Ross

2021年9月高级计算机网络



Computer
Networking: A Top
Down Approach
6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012

Multimedia Networking 2-1

多媒体网络：提纲

- 2.1 多媒体网络应用
- 2.2 流式存储视频
- 2.3 voice-over-IP
- 2.4 实时交互式应用的协议
- 2.5 多媒体的网络支持

Multimedia Networking 2-3

多媒体网络

❖ 互联网多媒体应用的现状

- VOD: Netflix, Hulu, 腾讯视频, 爱奇艺, B站, 优酷土豆, 芒果
- Youtube, 抖音: 多媒体内容的消费者和生产者
- 虎牙、斗鱼、快手: 直播平台
- 互联网电话和视频聊天(支持多人)应用: Skype, Google Talk, zoom、腾讯会议、微信、钉钉和 QQ(中国)

❖ 趋势预测

- 未来若干年, 所有的视频分发和视频会议都会通过互联网开展
- 条件: 高速无线接入(wifi, 4G, 5G)

Multimedia Networking 2-2

多媒体网络：提纲

- 2.1 多媒体网络应用
- 2.2 流式存储视频
- 2.3 voice-over-IP
- 2.4 实时交互式应用的协议
- 2.5 多媒体的网络支持

Multimedia Networking 2-4

2.1 多媒体网络应用

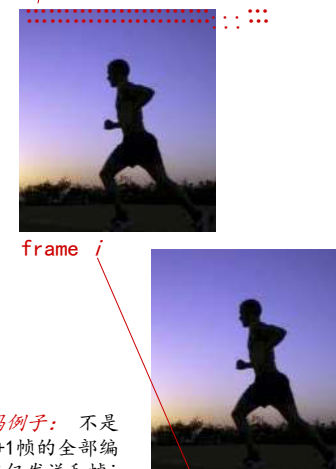
- 多媒体网络应用：涉及到音视频等多媒体的网络应用
- 本节主要内容
 - 网络多媒体应用分类
 - 对网络服务的需求
 - 带来的设计问题

Multimedia Networking 2-5

多媒体：视频

- ❖ 视频：固定速度显示图像序列
 - e. g. 24 images/sec
- ❖ 数字化图像：像素的阵列
 - 每个像素被若干bits表示
- ❖ 编码：使用图像内和图像间的冗余来降低编码的比特数
 - 空间冗余(图像内)
 - 时间冗余(相邻的图像间)
- ❖ 网络视频特点：
 - ❖ 高码率：>10x 音频, 高带宽需求
 - ❖ 可以被压缩
 - ❖ 90%以上的网络流量是视频

空间编码例子：不是发送N个相同的颜色（全部是紫色）值，仅仅发送2各值：颜色（紫色）和重复的个数（N）

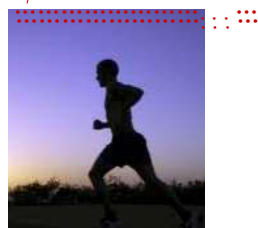


Multimedia Networking 2-6

多媒体：视频

- ❖ CBR: (constant bit rate): 以固定速率编码
- ❖ VBR: (variable bit rate): 视频编码速率随时间的变化而变化
- ❖ 例子:
 - MPEG系列
 - MPEG 1 (CD-ROM) 1.5 Mbps
 - MPEG2 (DVD) 3-6 Mbps
 - MPEG4 (often used in Internet, < 1 Mbps)
 - H.26X序列
 - AVS
 - ...

空间编码例子：不是发送N个相同的颜色（全部是紫色）值，仅仅发送2各值：颜色（紫色）和重复的个数（N）



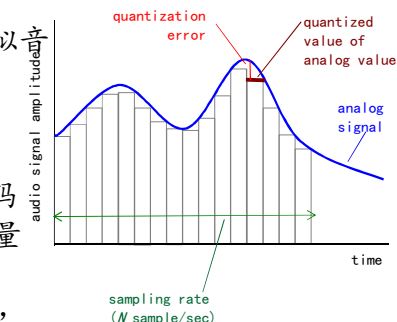
时间编码例子：不是发送第i+1帧的全部编码，而仅仅发送和帧i差别的地方



Multimedia Networking 2-7

多媒体：音频

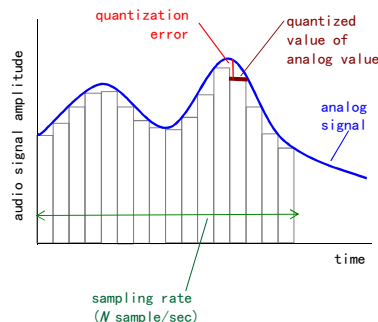
- ❖ 非压缩编码的PCM过程：采样、量化和编码
- ❖ 采样：以固定采用频率采样模拟音频信号
 - 电话：8000采样/sec
 - CD音乐：44,100采样/sec
- ❖ 量化编码：每个采用被量化编码
 - 如：2⁸=256可能的量化值，量化误差
 - 每个量化值被若干比特表示，如：8 bits
- ❖ 音频：码率低
 - 人对音频的质量比较敏感



Multimedia Networking 2-8

多媒体：音频

- ❖ 例子：8,000采样/秒，256量化级别：64,000bps
- ❖ 接收端将比特转换成模拟信号：
 - 存在量化误差
 - 一些质量损失
- 音频的**压缩编码**：mp3, AAC, GSM



Multimedia Networking 2-9

速率的例子

- ❖ CD: 1.411 Mbps
- ❖ MP3: 96, 128, 160 kbps
- ❖ 互联网电话: 5.3 kbps及以上

多媒体网络：3种应用类型

1. 流化、存储音视频

- 存储（在服务器、非实时产生的数据）：
 - 数据以前生产并存储
 - 可以按照比正常音视频播放所需速度更快的速度传输，意味着可在客户端进行缓存/预取
- 流化（单向）：在下载整个文件前就可以播放
 - 网络带宽够且服务器已经存储（相反实时音视频不可以）
- e.g., YouTube, Netflix, Hulu
- 流化存储音视频特点：流化、交互性、连续播放
- 对网络的需求：平均带宽 > 播放带宽，延迟10s可接受
 - 路径带宽变化，缓存平滑吞吐的变化，但本质上从长期来看，路径平均吞吐 > 播放所需要带宽
- 所使用的技术：缓存、预取、自适应带宽质量和CDN

Multimedia Networking 2-10

多媒体网络：3种应用类型（续）

2. IP交互式音频/视频

- 交互式音频：互联网电话VoIP
- 交互式视频：Skype，微信，Google Talk，
- 特点：数据**实时**产生、流化、**双（多）向**
- 对网络的需求：
 - 有带宽要求，延迟要小150ms，人-人交互特性限制了延迟容忍度
 - 抖动要小，对于丢失不是很敏感
- 使用的技术：自适应播放，FEC，错误掩盖

Multimedia Networking 2-11

多媒体网络：3种应用类型（续）

3. 流化、实时音视频

- 网络视频直播、网络广播
 - e.g., 实况赛事(足球)
- 2种实现方式：网络层组播、应用层组播
- 数据**实时**产生，流化，**单向(用户多)**
 - 直播平台支持交互
- 对网络的要求：带宽要求，延迟10s可接受
- 使用技术和流式存储视频技术类似
 - 初始缓存、自适应带宽、CDN分发

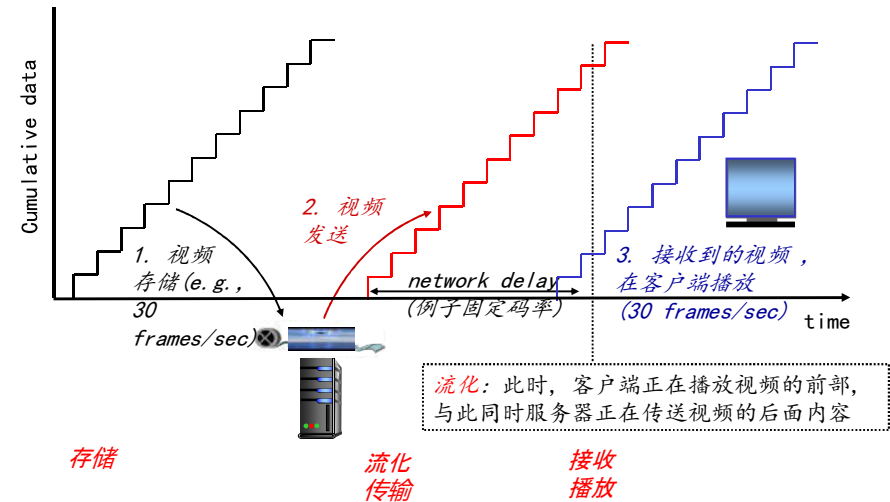
Multimedia Networking 2-12

多媒体网络：提纲

- 2.1 多媒体网络应用
- 2.2 流式存储视频
- 2.3 voice-over-IP
- 2.4 实时交互式应用的协议
- 2.5 多媒体的网络支持

Multimedia Networking 2-13

流式存储视频



Multimedia Networking 2-14

中科大高网-21秋

流式存储视频：挑战

- ❖ **连续播放限制**：一旦客户端开始播放，消费速率=原始速度
 - 到点该播放某帧了，该帧的数据就应该收到
 - 否则以后到了也没有用
 - 但**网络延迟是变化的** (抖动)，有可能某些帧延后到，因此需要客户端**缓存**
 - 第一个帧收到不是立即放，而是延迟一段时间再放
- ❖ 其它挑战：
 - 客户交互行为：暂停、前进、倒退、在视频内部跳转
 - 视频分组可能丢失，重传（需要时间代价）

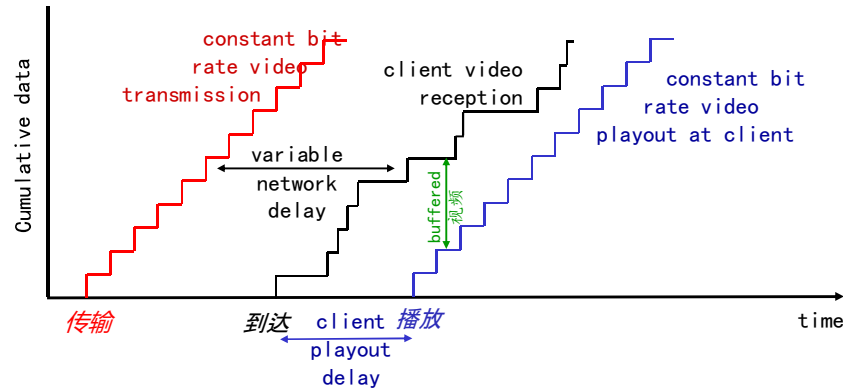
Multimedia Networking 2-15

2.2 流式存储视频

- ❖ **缓冲的作用**：
 - 吸收端到端延迟的变化：某帧延后到达，只要缓冲中还有数据，就可以连续播放；靠增加总体播放延迟作为代价来弥补延迟变化
 - 补偿端到端带宽的波动：瞬间路径吞吐<播放码率，进来的速率<消耗的速率，只要缓冲中还有数据，就可以连续播放；条件：路径平均吞吐>=播放带宽
- ❑ **流化视频技术**
 - UDP流化
 - HTTP流化
 - 自适应HTTP流化
- ❖ **都需要客户端缓冲**：因为网络延时是变化的，路径平均吞吐是变化的

Multimedia Networking 2-16

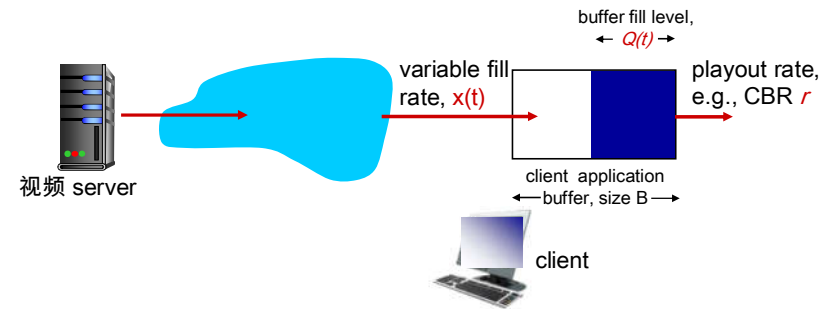
流式存储视频：缓存



- ❖ 客户端缓存和播放延迟：补偿网络增加的延迟，以及延迟抖动
- ❖ 增加播放延迟换取播放的连续性

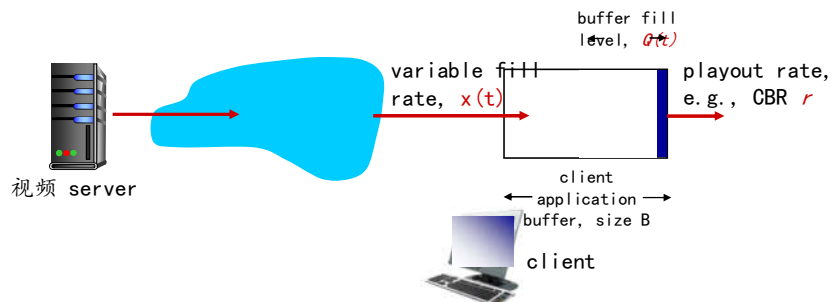
Multimedia Networking 2-17

客户端缓存、播放



Multimedia Networking 2-18

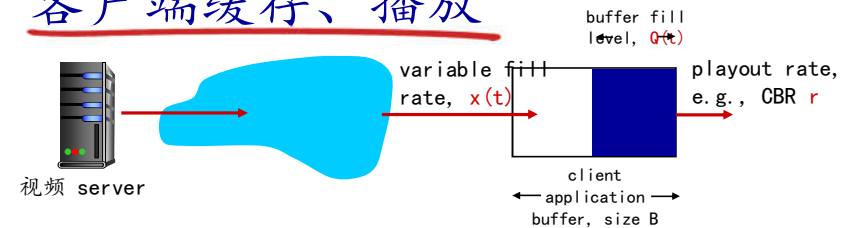
客户端缓存、播放



1. 开始注入缓存，直到播放开始时间 t_p 到
2. 播放开始于 t_p
3. Buffer缓冲的数据量随着时间变化：注入速率 $x(t)$ 是变化的，播放速率是固定的

Multimedia Networking 2-19

客户端缓存、播放



播放缓存：平均注入速率 x ，播放速率 r

- ❖ $x < r$ ：缓存最终变空(导致视频播放停顿直到缓存重新获得注入)
- ❖ $x > r$ ：缓存不会空，假设初始播放延迟足够大，可以吸收 $x(t)$ 的变化(如服务器按照 r 发送，则不会长时间大于 r)
- ❖ 初始播放延迟的权衡：
 - 大的播放延迟，缓存变空播放停顿的可能性小
 - 极致是下载完毕再播放，非流化
 - 但是引入延迟大，用户等待时间长

Multimedia Networking 2-20

流式多媒体技术1: UDP

- ❖ 服务器以恒定速率向客户端发送数据
 - 通常: 发送速率= 编码速率 = 恒定速率
 - 由于拥塞控制状况, 达到接收端的速率会变化
- ❖ 缓存: 引入一个小的播放延迟(2-5s)用于平滑网络抖动
- ❖ 错误恢复: 在应用层做, 如果时间允许
- ❖ RTP [RFC 2326]: 封装所需要的多媒体类型、时戳、序号: RTP→UDP
- ❖ RTSP: 与RTP配合, 让客户和服务端交互一些控制命令: 暂停、前进等
- ❖ UDP可能不能穿过防火墙

Multimedia Networking 2-21

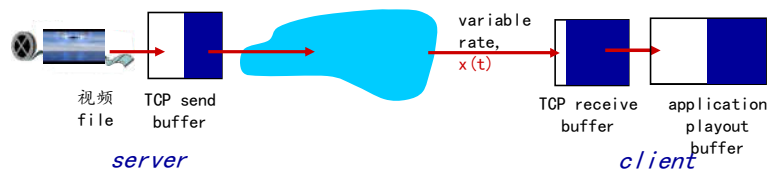
流式多媒体技术1: UDP (续)

- ❖ UDP流化的思路:
 - ❖ 网络带宽、抖动情况不可控
 - ❖ 尽量在源端保持恒定速率(无流量控制和拥塞控制)
- ❖ UDP流化的问题:
 - ❖ 客户端-服务器之间的带宽变化, 且不可预测, 恒定码率的UDP流化不能够提供连续播放
 - ❖ 需要一个控制协议, RTSP服务器, 增加了部署大规模VOD服务器的代价
 - ❖ 很多防火墙阻止UDP分组

Multimedia Networking 2-22

流式多媒体技术2: HTTP

- ❖ 部署和运行
 - 服务器把视频文件当成HTTP对象, 有url对应;
 - 客户端请求, 建立TCP连接, 发出HTTP GET;
 - 服务器在拥塞控制、流量控制条件下, 尽可能快地将文件发送给客户端;
 - 速率是变化的: 拥塞控制、流量控制和重传机制
 - 客户端: 客户应用层缓冲; 达到门限(一定量, 播放延迟 t_p), 开始播放;



Multimedia Networking 2-23

流式多媒体技术2: HTTP

- ❖ 之前的思路: TCP不太适合作多媒体流化服务
 - 网络通路上带宽和延迟情况, 主机无法控制, 在源端做好
 - 引入TCP波动叠加: 网络本身+引入TCP带来的额外波动
 - 拥塞控制、流量控制和重传机制
 - 应用层的速率(编码速率)和向网络注入的速率很不一致
 - 还没有从主机出去, 码率已经乱了, 流化效果不好
- ❖ http流化的思路:
 - 本质上: 通路上平均带宽 > 播放码率
 - 引入TCP叠加的更大波动, 但是采用缓冲、预取等技术进行, 也是可行的, 可以连续播放(引入大的播放延迟)
 - 利用TCP好处: 在带宽较大情况下, 尽可能地传到客户端
 - 在客户端缓冲满的情况下, 反向压力, 让发送端暂时不发送, 也不会溢出

Multimedia Networking 2-24

流式多媒体技术2: HTTP

- ❖ 代价:
 - 应用层需引入更大的播放延迟、来平滑TCP交付速率的变化
 - 原因: 网络本身的抖动带宽变化, 以及TCP本身流量控制、拥塞控制引入额外的抖动、带宽变化, 叠加了波动
- ❖ 好处:
 - HTTP/TCP更容易通过防火墙、NAT
 - 无需RTSP服务
- ❖ 适合存储数据的流化、不适合实时数据的流化
 - 坏处: 采用TCP引入更多的延迟变化, 影响交互性和实时性, 不适合实时数据的流化
 - 无法利用http/TCP的可以预取的好处:
 - 无法预期还未生成的实时数据

Multimedia Networking 2-25

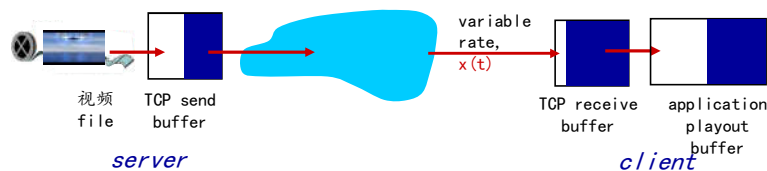
流式多媒体技术2: HTTP

- ❖ 缓存: 接收到一定数量、然后客户端按照播放码率进行播放
- ❖ 预取:
 - 客户端按照高于消费速率 (尽可能快地) 来预取数据, 为以后播放做准备;
 - 预取天然地和TCP捆绑在一起
- ❖ 例子:
 - 网络视频的播放带宽1Mbps, 网络的可用带宽1.5Mbps
 - 客户端按照1.5mbps的速率请求和存储速率
 - 启动播放延迟小, 而且预存以后消费的数据支持连续播放
 - 在应用缓冲满的时候, 客户端就不请求了

Multimedia Networking 2-26

流式多媒体技术2: HTTP

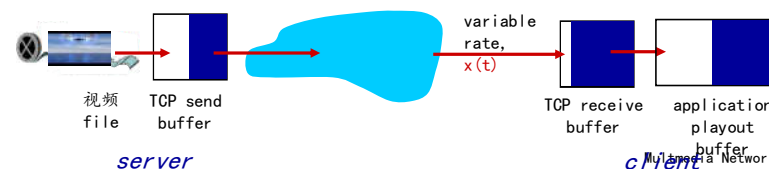
- ❖ 客户端的2级缓存:
 - TCP缓存+应用层缓存
- ❖ 讨论1: 应用缓存=文件大小, 非流化
- ❖ 讨论2: 暂停播放: 压力反向传递
 - 应用不从缓存取数据, 应用缓存满, 阻止TCP往上交, 阻止服务器TCP传, 服务器缓存满, 服务器应用停止传
- ❖ 讨论3: 正常播放, 但是 $x >$ 播放速率
 - 应用缓存满, TCP缓存满, 服务器TCP缓存满, 服务器不传



Multimedia Networking 2-27

流式多媒体技术2: HTTP

- ❖ 场景描述:
 - B: 客户端应用层缓存大小;
 - Q: 存下Qbits后才开始播放, 初始播放延迟 ($Q < B$);
 - r: 视频消费速率, 例如: 3Mbps;
 - x: 在播放时, 服务器发送速率是 x (假设是恒定)
 - $t=0$, 缓冲为空
- ❖ 开始播放时间: $t_p = Q/x$



Multimedia Networking 2-28

流式多媒体技术2: HTTP

❖ $t=?$, 缓冲满

■ 情况1: $x < r$, 网络速度小于消费速度

- 缓冲 t_p 开始播放, 很快被消费完毕(服务器发送速度低), 停顿
- t_p 时间再开始播放
- 结论: 播放在(连续播放和停顿)之间循环
- 永远满不了

■ 情况2: $x > r$,

- T_p 时间开始播放
- 然后在 $(B-Q) / (x-r)$ 时间内满;
- 发送停顿, 播放连续
- 结论: 网络速度大于消费速度, 播放可以连续进行;

Multimedia Networking 2-29

流式多媒体技术2: HTTP

❖ 暂停播放: 数据源源不断的送过来, 缓冲满了, 停止发送

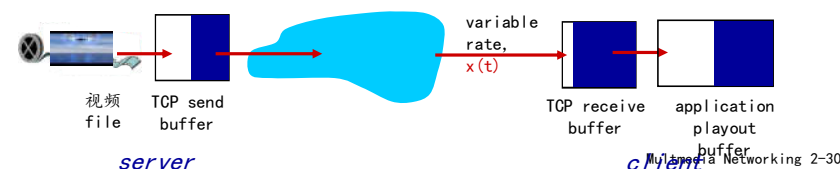
❖ 重新定位:

- HTTP GET 头部指定 请求视频文件的范围
- 如果重新定位的时间: $t > t_0 + B/r$: 只要重新定位在此时间之后, 预取的数据无用

❖ 放弃播放: 缓存和预取的数据全部无用

❖ 对策:

- 适中的应用层缓存大小;
- 预取数据的大小要限制;



Multimedia Networking 2-30

流式多媒体技术3: DASH

❖ HTTP流化的问题

- ❖ 所有的客户端不管网络状况如何都请求同一个码流

❖ DASH: Dynamic, Adaptive Streaming over HTTP

❖ 服务器:

- 将视频文件分割成多个块
- 每个块独立存储, 编码于不同码率(8-10种)
- 货物清单 (manifest file): 提供不同块的URL

Multimedia Networking 2-31

流式多媒体技术3: DASH

❖ 客户端:

- 先获取货物清单
- 周期性地测量服务器到客户端的带宽
- 查询货物清单, 在一个时刻请求一个块, HTTP头部指定字节范围
 - 如果带宽足够, 选择最大码率的视频块
 - 会话中的不同时刻, 可以切换请求不同的编码块(取决于当时的可用带宽)

❖ DASH的优点

- 动态估计带宽情况, 当前缓存情况, DASH通常能够做到持续播放
- 减轻服务器的负担, 可扩展性强

Multimedia Networking 2-32

流式多媒体技术3: DASH

- ❖ “智能”客户端: 客户端自适应决定
 - 什么时候去请求块 (不至于缓存挨饿, 或者溢出)
 - 请求什么编码速率的视频块 (当带宽够用时, 请求高质量的视频块)
 - 哪里去请求块 (可以向离自己近的服务器发送URL, 或者向高可用带宽的服务器请求)

Multimedia Networking 2-33

Content Distribution Networks

- ❖ 挑战: 服务器如何通过网络向上百万用户同时流化视频内容 (上百万视频内容)?
 - ❖ 选择1: 单个的、大的超级服务中心 “mega-server”
 - 服务器到客户端路径上跳数较多, 瓶颈链路的带宽小导致停顿
 - “二八规律”决定了网络同时充斥着同一个视频的多个拷贝, 效率低 (付费高、带宽浪费、效果差)
 - 单点故障点, 性能瓶颈
 - 周边网络的拥塞
- 评述: 相当简单, 但是这个方法不可扩展

Multimedia Networking 2-34

Content distribution networks

- ❖ 选项2: 通过CDN, 全网部署缓存节点, 存储服务内容, 就近为用户提供服务, 提高用户体验
 - enter deep: 将CDN服务器深入到许多接入网
 - 更接近用户, 数量多, 离用户近, 管理困难
 - Akamai, 1700个位置
 - bring home: 部署在少数(10个左右)关键位置, 如将服务器簇安装于POP附近 (离若干1stISP POP较近)
 - 采用租用线路将服务器簇连接起来
 - Limelight

Multimedia Networking 2-35

CDN内容复制策略

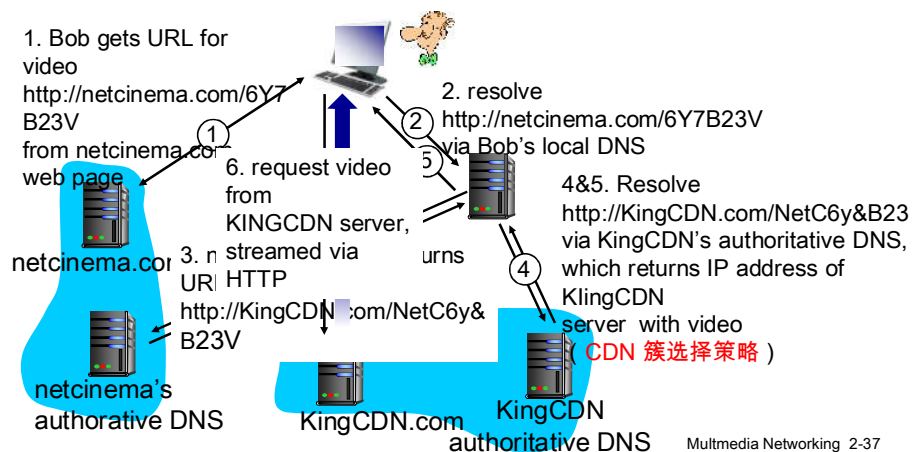
- ❖ 不需要将所有内容复制到所有的缓存节点中
- ❖ 内容复制策略决定什么内容存储在缓存节点中
 - 内容+缓存节点
- ❖ 非热点内容会被替换 (替换策略) 掉, 以腾出空间
 - 在某个缓存节点上, 内容随时间变化关系

Multimedia Networking 2-36

CDN: “简单” 内容访问场景

Bob (客户端) 请求视频 `http://netcinema.com/6Y7B23V`

- 视频存储在CDN, 位于 `http://KingCDN.com/NetC6y&B23V`



CDN簇选择策略

❖ 挑战: CDN DNS如何选择“好”的CDN节点, 为客户提供流化服务

- 选择离客户物理位置较近的CDN节点:
 - CDN运营商维护地理数据库
 - 问题: 地理上近的不见得是网络上近的, 没有考虑网络波动
- 采用cdn服务器簇到客户端 网络情况较好的簇
 - 很多具体方法
 - 如: CDN 节点定期ping接入网络ISP, 报告给CDN DNS

CDN簇选择策略

❖ 挑战: CDN DNS如何选择“好”的CDN节点, 为客户提供流化服务

- IP任意播anycast
 - CDN服务器簇在一个anycast组中, 有个组地址
 - 每个CDN服务器簇向外做标准的任意播组地址的通告
 - BGP路由器收到多个属于一个组的通告, 保留一个较好的 (AS跳数较少的) 路径
 - 使用阶段: 客户希望看某个视频, CDN域名系统返回任意播地址; 分组被路由到最近的cdn服务器簇;

CDN簇选择策略

❖ 另外一个选择: 让客户端决定-给客户端一个CDN服务器的列表

- 客户端ping服务器, 选择最好的
- Netflix方法

❖ 簇选择策略要考虑的因素:

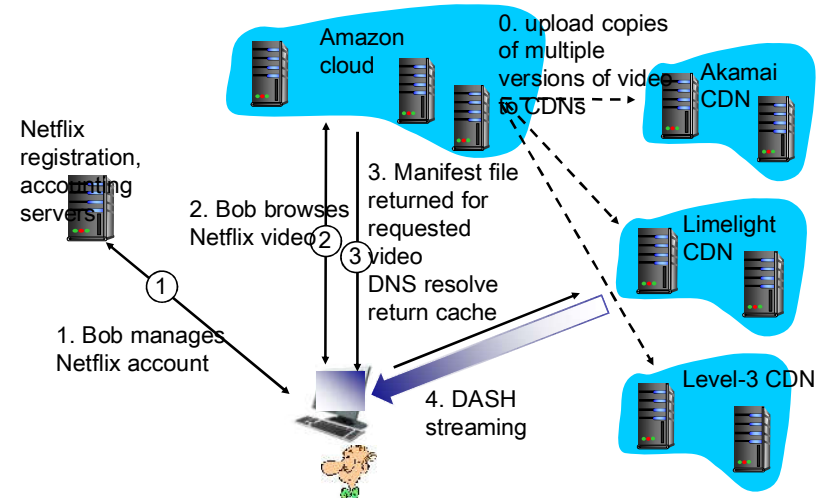
- 在内容在某些CDN簇的前提下
- CDN服务器簇到客户的网络状况: 延迟, 带宽, 丢失;
- 还要考虑负载: 不要定向到负载较重的服务器上;
- ISP交付代价问题 (流量合约问题)

案例：Netflix

- ❖ 2011年的30% US下载流量
- ❖ 自己拥有非常少的基础设施站点，大量地采用了第三方服务器提供的服务：
 - 自己：注册和付费服务器
 - Amazon (3rd party) 云服务
 - Netflix 将内容上载到Amazon云中，内容存储
 - 在云中创建切块 (4s)，转换成不同码率的版本
 - 将多个版本从云上载到CDN中，便于DASH
 - 业务web网页存在云中，用于用户的浏览
 - 3个第三方CDN存储和流化：Akamai, Limelight, Level-3

Multimedia Networking 2-41

案例学习：Netflix



Multimedia Networking 2-42

案例：YouTube

- ❖ 谷歌旗下，全球最大视频平台和服务商：上百万视频/天，5亿用户点播每天；
- ❖ 特点：不采用第三方服务，全部是自己的基础设施
- ❖ 技术：私有协议
 - 采用google自己的CDN分发视频内容；google采用了大概50个cdn簇存储和分发youtube内容；
 - 采用DNS重定向选择向客户提供服务的CDN服务器簇；
 - 选择策略：RTT最小；
 - 为了负载均衡，可能让较远的CDN服务器簇向用户提供服务
 - 采用HTTP流化，到2011年不支持DASH，
 - 为了解决带宽和服务器资源，在客户端预取了一些数据之后，限制HTTP get字节的多少
 - 上下下载都采用HTTP

Multimedia Networking 2-43

案例：kankan等

- ❖ 以上的视频内容运营商方案，部署服务器簇，租用带宽等，费用高
- ❖ 另外一种思路：p2p方式部署：Kankan, pptv, pps
- ❖ 不是由服务器群向客户提供服务，而是客户之间相互服务
- ❖ VOD业务：客户端点播某个影片，首先需要从tracking server那里获得一个组成员信息，它们之间交换影片的map文件，相互服务
 - 客户存储部分内容，可以就这些内容向其他客户端提供流化服务
- ❖ 直播业务：
 - 在服务器的指挥下，构建组播树
 - 客户端一边接流，一边向其他若干客户端提供流化服务

Multimedia Networking 2-44

多媒体网络：提纲

- 2.1 多媒体网络应用
- 2.2 流式存储视频
- 2.3 voice-over-IP
- 2.4 实时交互式应用的协议
- 2.5 多媒体的网络支持

Multimedia Networking 2-45

Voice-over-IP (VoIP)

- ❖ VoIP: 网络上传输实时产生的音频, 应用需求h:
 - 延迟不能够太大, <400ms否则会影响交互性-
 - 应用处理不能够增加太多延迟, 采用UDP (传输层不加太多延迟)
 - 少量丢失允许, 但如丢失太多, 会听不清
 - EFC
 - 抖动太大, 变调
 - 需要缓冲处理, 持续播放; 需要编号排序
- ❖ IP提供的服务: 尽力而为
 - 丢失; 延迟; 抖动
- ❖ 在应用层上做哪些工作, 对抗网络的问题, 从而能够很好地运行VoIP
 - 都是IP分组, 网络层不单独为某种类型应用优化

Multimedia Networking 2-46

中科大高网-21秋

Voice-over-IP (VoIP)

- ❖ 问题1: 丢失
 - 缓冲区满、出错校验没有通过
 - 延迟超大 (超过人工播放延迟), 错过了播放时间到达
- ❖ 采用TCP不可行
 - 超时重发机制会让整体增加延迟, 影响实时性
 - 无法利用TCP尽力传输后面产生的数据, 数据实时产生
- ❖ 在UDP上传输实时产生的音频
- ❖ 应用层利用编码恢复或者掩盖部分丢失, 小于1-10%的丢失都没有关系

Multimedia Networking 2-47

中科大高网-21秋

Voice-over-IP (VoIP)

- ❖ 问题2: 延迟
 - 所有延迟: 各链路的传输, 传播延迟, 应用层缓冲和处理等
- ❖ 策略是: 无法减少必要的网络延迟, 在应用层不能够人工增加延迟: 处理, 缓冲
 - 采用UDP, 而不是采用TCP
 - 引入的播放延迟, 使得总体延迟最好限制在400ms以内
 - 在保证最大化连续播放情况下, 最小化播放延迟

Multimedia Networking 2-48

Voice-over-IP (VoIP)

❖ 问题3: 抖动

- 原因: 分组延迟不同, 不同路径, 有的链路重传
- 后果: 分组之间延迟有差, 分组间间隔变化; 甚至先发后到→直接播放效果差 (变调, 不可理解)

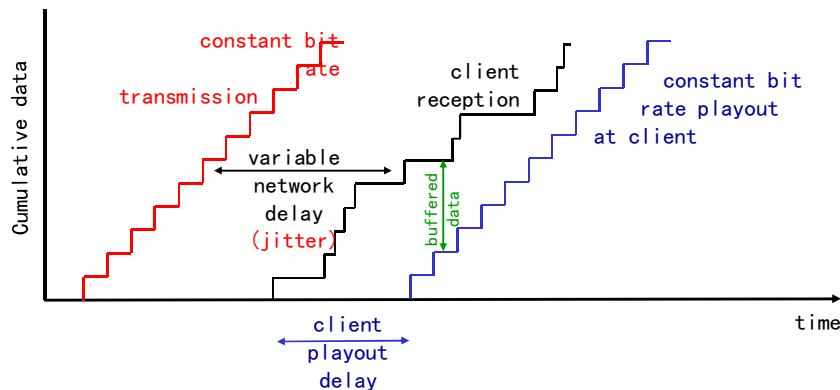
❖ 措施

- 发送端分组打时戳和编号, 接收端排序
 - 序号+1, 序号在静默期不发数据时不增加
 - 时戳+, 时戳在静默期仍然增加
- 在接收端引入缓冲, 引入播放延迟消除 (吸收) 抖动
 - 固定或者自适应的

Multimedia Networking 2-49

Multimedia Networking 2-50

延迟抖动



- ❖ 2个连续分组的端到端延迟差: 可以超过或者小于20ms (分组经历了不同的延迟)

Multimedia Networking 2-51

VoIP架构

❖ 说话者的音频: 分成突发期和静默期

- 会话突发期: 64 kbps
- 只是在会话突发期产生分组
- 20msec生产一个数据块, 8000B/sec: 160B/20msec

❖ 在每一块语音数据上加上应用层报头

❖ 应用报文 (头+语音数据) 封装在UDP或TCP段中

❖ 在会话突发期, 应用程序每隔20ms发送一个段

VoIP: 固定播放延迟

❖ 接收方试图在每个数据块产生qms之后播放它, 人工引入一些延迟平滑抖动

- 数据块时戳: 在 $t+q$ 播放它
- 如果数据块到达晚于 $t+q$: 数据到来太晚了, 数据“丢失”了

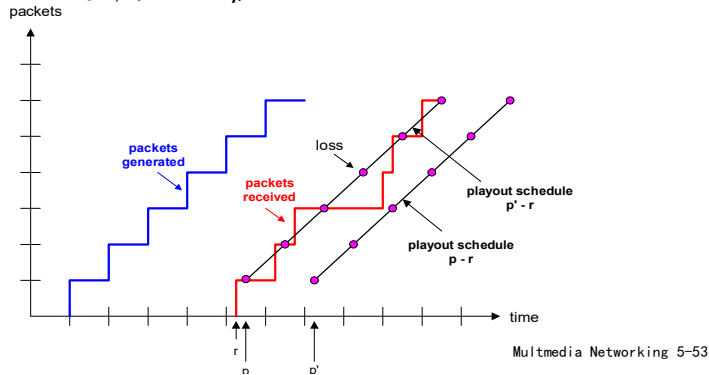
❖ 选择q的权衡

- 大的q: 较少的分组丢失, 但延迟大, 影响交互性
- 小的q: 好的交互体验, 但是丢失分组较多

Multimedia Networking 2-52

VoIP: 固定播放延迟

- 语音突发期，发送方每20ms产生分组，第一个分组在 r 时刻接收
- 方案1: 播放时间开始于 p
- 方案2: 播放的时间开始于 p'
- 固定播放延迟: 不能够适用于所有网络(延迟和抖动情况不一样, 无法选择合适的 q)



自适应播放延迟 (1)

- ❖ 目标: 尽可能小的播放延迟, 同时要小的丢失率
- ❖ 方法: 自适应播放延迟调整
 - 估计网络延迟, 在每个语音突发期开始调整播放延迟
 - 静默期实际上被压缩或者扩张了
 - 在突发期数据块一旦播放, 数据块仍然是每个20ms播放一个
- ❖ 自适应估计分组延迟: (EWMA - exponentially weighted moving average, recall TCP RTT estimate):

$$d_i = (1-a)d_{i-1} + a(r_i - t_i)$$

d_i : delay estimate after i th packet
 a : small constant, e.g. 0.1
 r_i : time received
 t_i : time sent (timestamp)
 $r_i - t_i$: measured delay of i th packet

Multimedia Networking 2-54

自适应播放延迟 (2)

- ❖ 也用于估计平均延迟的偏差 v_i :

$$v_i = (1-b)v_{i-1} + b|r_i - t_i - d_i|$$
- ❖ 对于每个接收到的分组迭代地估计 d_i , v_i , 但只是在下一个突发期使用该值
- ❖ 对这个突发期的第一个分组, 播放时间为:

$$\text{playout-time}_i = t_i + d_i + Kv_i$$

- 突发期的后续分组都在在此播放延迟之后, 每隔20ms播放一块, 周期性播放
- K默认为4

自适应播放延迟 (3)

- Q: 接收端如何判断突发期的第一个分组到来了(新的突发期开始了)?
- ❖ 序号: 源端有数据发送, 每个分组序号+1
- ❖ 时间戳: 不管有无数据发送, 时间戳都向前走
 - ❖ 前例: 每125us走一个单位
- ❖ 判断突发期, 没有丢失情况; 接收方将会看到连续时戳
 - 序号连续
 - 时戳差 > 20 msec (>160)
- ❖ 判断突发期, 发生丢失情况: 接收方必须检查时戳和序号两个字段
 - 连续分组的时戳差 > 20 msec
 - 序号差2个或以上
 - 时戳差20ms(160), 序号差1
 - 时戳差40ms(320), 序号差2
 - 时戳差60ms(480), 序号差3

VoiP: 从分组丢失中恢复 (1)

挑战: 从分组丢失中恢复, 在给定小的可容忍延迟内 (分组开始发送到该分组应该播放之间 $t \sim t+q$)

- ❖ 利用重传: 每个ACK/NAK 将会耗时 ~ 1 RTT, 不可行
- ❖ 其他选择: **Forward Error Correction (FEC)**
 - 发送足够的bits允许从丢失中恢复, 无需重传

简单FEC

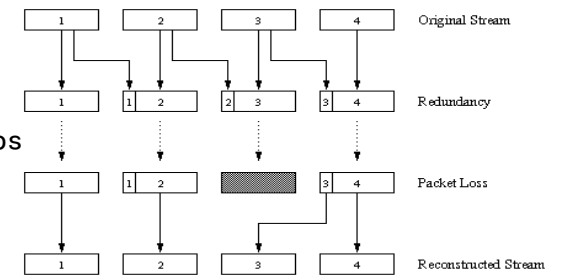
- ❖ 对于每组n个数据块, 用n块数据的异或创建一个冗余块
- ❖ 发送n+1数据块, 增加带宽约合 $1/n$
- ❖ 至多一组中一块数据丢失的情况下, 可以重构该丢失的数据块, 无需重传
 - 代价: 延迟

Multimedia Networking 2-57

VoiP: 从分组丢失中恢复 (2)

另外一个FEC方案:

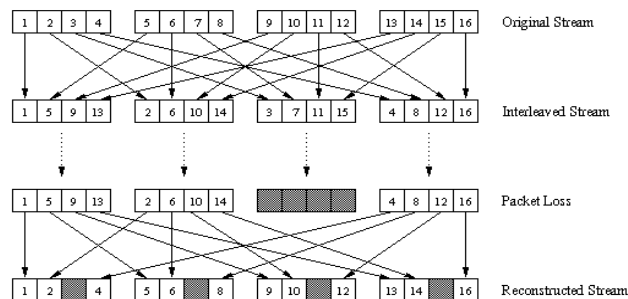
- ❖ 捎带低质量码流
- ❖ 将低质量码率当成冗余信息传递
- ❖ e.g., PCM码流64 kbps, 冗余码率GSM速率13 kbps



- ❖ 非连续的丢失: 接收方可以掩盖丢失
- ❖ 一般化方法: 也可以附加 (n-1)st 和 (n-2)nd 低码率数据块

Multimedia Networking 2-58

VoiP: 从分组丢失中恢复 (3)



交织掩盖丢失:

- ❖ 音频数据块分割成小的单元, e.g. 4个5 msec数据块 (每20ms数据块)
- ❖ 分组包含不同数据块的小单元
- ❖ 如果分组丢失, 仍然拥有每个原始数据块的绝大多数小单元
- ❖ 没有冗余信息, 但增加了播放延迟

Multimedia Networking 2-59

Voice-over-IP: Skype

- ❖ 私有应用层协议 (通过反向工程推断)

- 报文加密

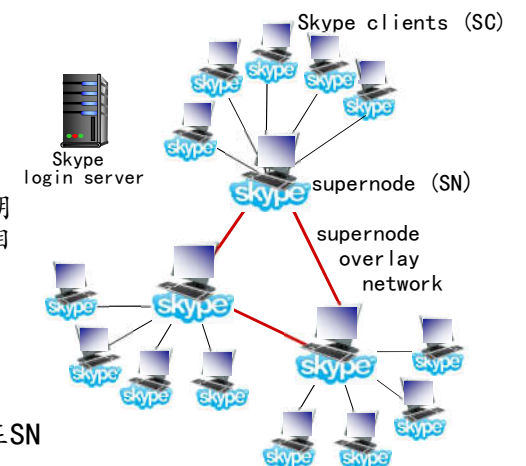
- ❖ P2P 元件:

- **clients:** VoIP会话期间skype对等体直接相连

- **super nodes (SN):** 拥有特殊功能的skype 对等体

- **overlay network:** 在SN和SC之间

- **login server**

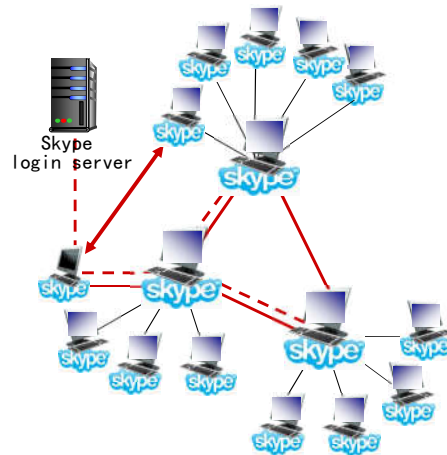


Application Layer 2-60

P2P voice-over-IP: skype

skype 客户端运行

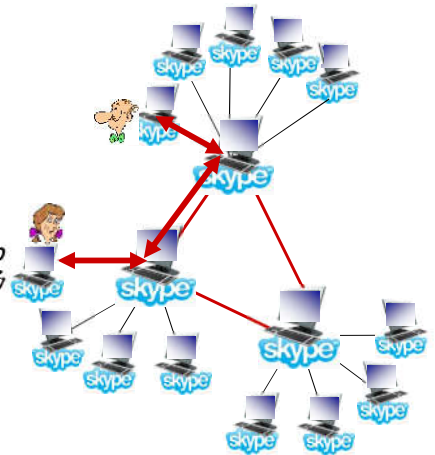
1. 采用TCP联系SN从而加入skype网络
2. 向集中式login服务器logs-in (username, password)
3. 从SN, SN覆盖网络获得被叫的IP地址,
 - 或者客户端好友的列表
4. 直接和被叫初始化会话



Application Layer 2-61

Skype: 对等节点当成中继

- ❖ **问题:** Alice和Bob都在NAT的后面
 - NAT阻止外部peer向内部peer初始化连接
 - 内部peer可以向外发起连接
- ❖ **中继解决方案:** Alice, Bob和他们的SN维护一个活跃的连接
 - Alice告诉SN她想连接Bob
 - Alice的SN连接Bob的SN
 - Bob的SN连接Bob (通过Bob和SN的连接)



Application Layer 2-62

多媒体网络: 提纲

- 2.1 多媒体网络应用
- 2.2 流式存储视频
- 2.3 voice-over-IP
- 2.4 实时交互式应用的协议
- 2.5 多媒体的网络支持

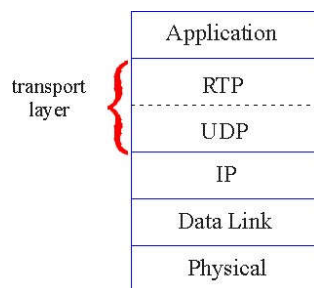
Real-Time Protocol (RTP)

- ❖ RTP 规范了用于携带音频和视频数据的分组结构
- ❖ RTP 运行在端系统
- ❖ RTP分组封装在UDP的段中
- ❖ RFC 3550
- ❖ RTP分组提供
 - 负载类型标示
 - 分组序号
 - 时戳
- ❖ 互操作性: 如果两个VoIP应用运行在RTP上, 他们可以协同工作

RTP 运行在UDP上

RTP 库提供传输层接口，扩展了UDP

- 端口号，IP地址
- 负载类型标示
- 分组序号
- 时戳



Multimedia Networking 5-65

RTP实例

例子: RTP在RTP上发送64kbps的PCM编码

- ❖ 应用程序收集编码数据，合成一个数据块，e. g. , 每20 msec = 160 B

- ❖ 音频数据块 + RTP 头形成RTP分组，封装在UDP段中

- ❖ RTP 头部标示了每个分组音频编码的类型
 - 发送方可以变换器编码类型，在会话期间
- ❖ RTP 头部也包括了序号和时戳

Multimedia Networking 2-66

RTP和QoS

中科大高网-21秋

- ❖ RTP 不提供任何按时交付以及其他QoS保证的机制
- ❖ RTP封装仅在端系统中能够看到（不会被中间的路由器看到，仅当成是一般的IP分组）
 - 路由器提供尽力而为的服务，没有确保RTP分组能够按时到达目标端的额外努力

RTP header

payload type	sequence number type	time stamp	Synchronization Source ID	Miscellaneous fields
--------------	----------------------	------------	---------------------------	----------------------

payload type (7 bits): 指示当前所用的编码类型，如果发送端在呼叫过程中变换其编码，发送端通过payload type字段指示接收方

Payload type 0: PCM mu-law, 64 kbps
Payload type 3: GSM, 13 kbps
Payload type 7: LPC, 2.4 kbps
Payload type 26: Motion JPEG
Payload type 31: H.261
Payload type 33: MPEG2 视频

sequence # (16 bits): 每传送一个RTP分组增加1

- ❖ 检测分组的丢失，重排分组次序

Multimedia Networking 2-67

Multimedia Networking 5-68

RTP header

payload type	sequence number type	time stamp	Synchronization Source ID	Miscellaneous fields
-----------------	-------------------------	------------	------------------------------	-------------------------

- ❖ **timestamp field (32 bits long)**: 反映了RTP第一个字节的采样时刻
 - 对于音频，每个采样周期，时戳增加1个 (e. g., 对于8KHz的采样，每隔125 usecs增加1)
 - 当源端活跃时每个RTP分组时戳增加160，应用程序产生160字节的编码后的采样数据，时戳增加160
 - 源端是非活跃状态时，时戳时钟也在持续增加
- ❖ **SSRC field (32 bits long)**: RTP流源的标示。每个RTP会话中的流都有一个不同的SSRC

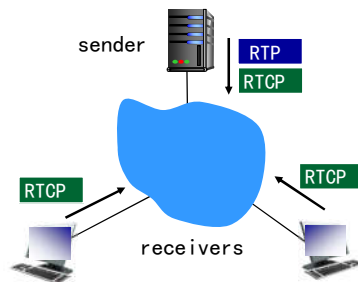
Multimedia Networking 2-69

Real-Time Control Protocol (RTCP)

- ❖ 与RTP协同工作
- ❖ 每个RTCP分组包含发送方和/或接收方的报告
 - 报告对于应用有用的统计信息: # packets sent, # packets lost, interarrival jitter
- ❖ RTP会话的每个参与方定期发送RTCP控制分组，到所有其他的参与方
- ❖ 反馈信息用于控制性能
 - 发送方可能会根据反馈调整其传输

Multimedia Networking 2-70

RTCP: 多个多播发送方



- ❖ 每个RTP会话: 通常有一个多播地址; 所有属于该会话的RTP/RTCP分组采用该地址
- ❖ RTP、RTCP分组采用不同的端口号相互区分
- ❖ 为了限制流量，在会话参与方增加时，每个参与方会降低其RTCP的流量

Multimedia Networking 5-71

RTCP: 分组类型

接收方报告分组:

- ❖ 分组丢失比例，最后序号，平均抖动

发送方报告分组:

- ❖ RTP流的SSRC，当前时间，已经发送分组的个数，已经发送的字节个数

源描述分组:

- ❖ 发送方e-mail地址，发送方名字，RTP流相关联的SSRC
- ❖ 提供SSRC和用户/主机名字的映射

Multimedia Networking 2-72

RTCP: 流同步

- ❖ RTCP可以在RTP会话里同步不同的媒体
- ❖ e. g., 视频会议应用: 每个发送方产生一个RTP视频流和一个音频流
- ❖ RTP分组中的时戳与视频相关联, 与音频采样时钟关联
 - 而不是关联到一个wall-clock时间
- ❖ 每个RTCP发送方报告分组包含 (与RTP关联的最近生成的分组):
 - RTP分组的时戳
 - 当分组生成时的wall-clock时间
- ❖ 接收方采用关联来同步视频和音频的播放

Multimedia Networking 2-73

中科大高网-21秋

SIP: Session Initiation Protocol [RFC 3261]

SIP 目标:

- ✓ 所有的电话呼叫, 视频会议都可以通过互联网展开
- ✓ 用户被名字或者email地址标示, 而不是电话号码
- ✓ 你可以接通被叫, 不管被叫是不是在漫游, 也不管被叫目前采用的IP地址是什么

Multimedia Networking 2-75

RTCP: 带宽扩展

RTCP 试图限制其流量不超过5%的会话带宽

例子: 一个发送方, 发送2 Mbps的视频

- ❖ RTCP试图限制其 RTCP 流量不差超过100 Kbps
- ❖ RTCP将其中的75%给所有的接收方; 将其中的 25%给发送方

- ❖ 75 kbps被均分给所有的接收方:
 - 有R接收方, 每个接收方获得 RTCP 流量: $75/R$ kbps.
- ❖ 发送方获得RTCP 流量: 25 kbps.
- ❖ 参与方决定通过计算平均RTCP分组大小/分配带宽, 来计算RTCP分组的周期

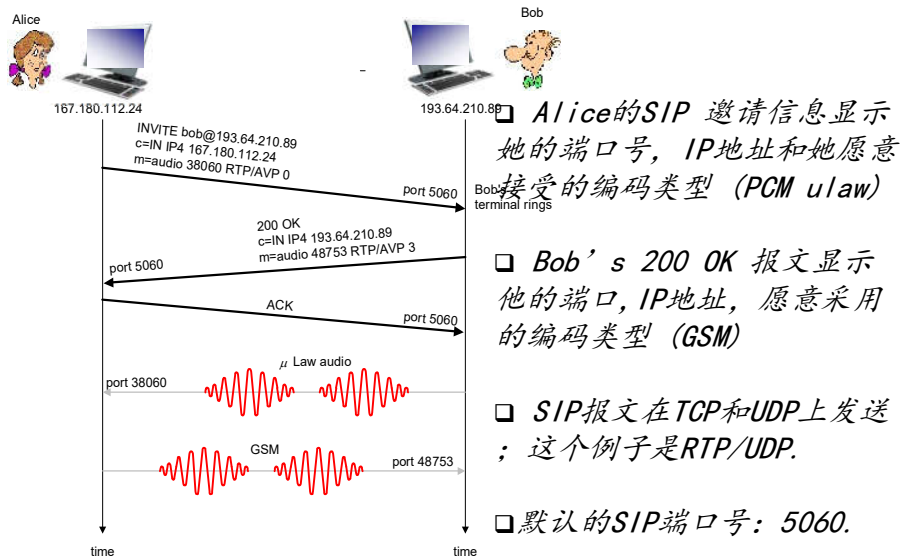
Multimedia Networking 2-74

SIP 服务

- ❖ 建立一个呼叫, SIP提供机制:
 - 对于主叫来说, 让被叫知道有人要和他建立呼叫
 - 主叫-被叫协商媒体类型和具体编码参数
 - 结束呼叫
- ❖ 确定被叫当前所使用的IP地址:
 - 将命名标示映射到当前被叫使用的IP地址
- ❖ 呼叫管理:
 - 在会话过程中加入一个新的媒体流类型
 - 在会话过程中改变编码
 - 邀请其他方加入会话
 - 转移、保持会话

Multimedia Networking 2-76

和一个已知IP地址建立一个呼叫



Multimedia Networking 5-77

建立呼叫(续)

- ❖ 编解码协商
 - 假设Bob没有PCM的编码器.
 - Bob用606 Not Acceptable应答, 列出他能够接收的编码类型, 之后Alice就可以在新的INVITE报文中建议采用新的编码
- ❖ 拒绝呼叫
 - Bob 可以拒绝呼叫, 应答是: “busy,” “gone,” “payment required,” “forbidden”
- ❖ 媒体数据可以通过RTP或者其它协议来传送

Multimedia Networking 2-78

SIP报文的例子

中科大高网-21秋

```
INVITE sip:bob@domain.com SIP/2.0
Via: SIP/2.0/UDP 167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
Call-ID: a2e3a@pigeon.hereway.com
Content-Type: application/sdp
Content-Length: 885

c=IN IP4 167.180.112.24
m=audio 38060 RTP/AVP 0
```

Notes:

- ❖ HTTP 报文语法
- ❖ sdp = session description protocol
- ❖ 每一个呼叫一个唯一的Call-ID

□ 这里我们不知道Bob的IP地址
需要一些中介SIP服务

- ❖ Alice 发送, 接收SIP报文, 采用默认的5060端口
- ❖ Alice 通过SIP客户端发送的头部信息规定, 通过UDP接收SIP消息receives SIP

Multimedia Networking 2-79

名字翻译和用户定位

- ❖ 主叫希望呼叫被叫, 但是仅仅知道被叫的名字或者email地址
- ❖ 希望得到被叫当前所使用主机的IP地址:
 - 用户移动
 - DHCP protocol
 - 用户使用不同的IP设备 (PC, PDA, car device)
- ❖ 呼叫的结果可能基于:
 - 一天中的时间 (work, home)
 - 主叫是谁 (不希望老板在你在家的时候呼叫你)
 - 被叫的状态 (calls sent to voicemail when callee is already talking to someone)

SIP服务器提供的服务:

- ❖ SIP registrar server
- ❖ SIP proxy server

Multimedia Networking 2-80

SIP registrar

- ❖ SIP server的一个功能: *registrar*
- ❖ 当Bob启动SIP客户端时, 客户端向Bob的注册服务器发送SIP注册报文

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
```

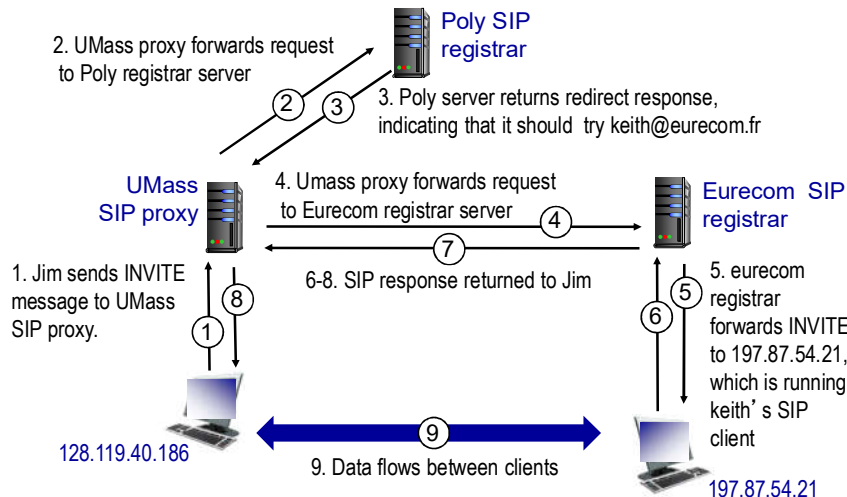
和BOB当前
Ip地址

和Bob沟通采用
UDP的5060端口

SIP proxy

- ❖ SIP服务器另外一个功能是: *proxy*
- ❖ Alice发送 invite报文给她自己的代理服务器
 - 包含地址: sip:bob@domain.com
 - 代理负责路由SIP报文到被叫, 可能需要经过多个代理
- ❖ Bob 经过相同SIP代理集合发送响应报文, 返回
- ❖ 代理得到Bob的 SIP响应报文, 返回给 Alice
 - 包含Bob的IP地址
- ❖ SIP代理类比于 local DNS server+TCP setup

SIP example: jim@umass.edu calls keith@poly.edu



和H. 323对比

- ❖ H. 323是另外一套为实时、交互式通信服务的信令协议
- ❖ H. 323来自ITU (telephony).
- ❖ H. 323是针对多媒体会议一个完整、垂直集成的协议族: signaling, registration, admission control, transport, codecs
- ❖ SIP来自IETF: 借鉴了很多HTTP协议的概念
 - SIP 协议具有 Web 风格,
 - H. 323具有电信风格
- ❖ SIP采用KISS原则: Keep it simple and stupid.
- ❖ SIP 是一个单独的协议, 需要和RTP协议配合, 但是也不一定, 也可以和其他的协议和服务组合使用

多媒体网络：提纲

- 2.1 多媒体网络应用
- 2.2 流式存储视频
- 2.3 voice-over-IP
- 2.4 实时交互式应用的协议
- 2.5 多媒体的网络支持

Multimedia Networking 2-85

应用层 vs 网络层 支持多媒体应用

- ❖ 前几节：在尽力而为网络上开展网络媒体服务的有效方法
 - 网络不做任何变化，在边缘端系统应用层做工作
 - 应用层技术：客户端缓存、预取、自适应媒体质量、自适应播放、丢失掩盖技术
 - 应用层系统级方法：CDN分发网络媒体内容
- ❖ 网络层机制：支持多媒体应用
 - 提供对网络多媒体有效支持

Multimedia Networking 2-86

中科大高网-21秋

网络层机制部署现状

- ❖ 还没有被广泛应用，不同ISP网络部署情况不同
- ❖ 原因1：应用程序或者应用基础设施+尽力而为服务：足够好
- ❖ 原因2：部署网络级的机制复杂，要对网络增加复杂的功能，不是很划算
 - 互联网的收费模式，让ISP不愿意投入太多
 - ISP投入多，ICP和服务运营商会受益更多，ISP很难分一杯羹
 - 单个ISP部署一些网络层的特性，但如果其他ISP没有部署，多媒体效果并不好（2017）

Multimedia Networking 2-87

多媒体的网络机制：3种

Approach	Granularity	Guarantee	Mechanisms	Complex	Deployed?
Making best of best effort service	All traffic treated equally	None or soft	No network support (all at application)	low	everywhere
Differentiated service	Traffic "class"	None or soft	Packet market, scheduling, policing	med	some
Per-connection QoS	Per-connection flow	Soft or hard after flow admitted	Packet market, scheduling, policing, call admission	high	little to none

方法1：最大限度地利用尽力而为的服务
应用层去做，网络层加带宽和容量
技术：Provisioning, dimensioning

Multimedia Networking 2-88

多媒体的网络机制：3种

Approach	Granularity	Guarantee	Mechanisms	Complex	Deployed?
Making best of best effort service	All traffic treated equally	None or soft	No network support (all at application)	low	everywhere
Differentiated service	Traffic "class"	None of soft	Packet market, scheduling, policing	med	some
Per-connection QoS	Per-connection flow	Soft or hard after flow admitted	Packet market, scheduling, policing, call admission	high	little to none

方法2: 区分服务 一类分组在路由器得到的服务 (例如: 得到优先权) 和另外一中类型不一样;
方法: 标记, 调度, 监管

Multimedia Networking 2-89

多媒体的网络机制：3种

Approach	Granularity	Guarantee	Mechanisms	Complex	Deployed?
Making best of best effort service	All traffic treated equally	None or soft	No network support (all at application)	low	everywhere
Differentiated service	Traffic "class"	None of soft	Packet market, scheduling, policing	med	some
Per-connection QoS	Per-connection flow	Soft or hard after flow admitted	Packet market, scheduling, policing, call admission	high	little to none

方法3: 每连接Qos保证, 建立时, 预约网络资源 (如带宽) 保证 端到端的性能
硬保证: 连接的性能确定得到保证;
软保证: 比较高的可能性得到性能保证
技术: 标记, 调度, 监管, 呼叫准入

Multimedia Networking 2-90

中科大高网-21秋

方法1: 最大限度利用尽力而为网络

- ❖ **分析:** 网络什么情况下对多媒体支持不好: 资源紧张拥塞时
- ❖ **思路:** 让资源不受限, 增加足够网络的资源 (带宽), 以至于让拥塞不发生、延迟小、抖动小
- ❖ **好处:** 无需改动尽力而为的网络服务
- ❖ **代价:** 带宽增容, 钱
- ❖ **方法:**
 - 给定网络拓扑, 估算一下链路的容量问题=> **带宽提供**;
 - 设计网络拓扑, 估算链路容量=> **网络估算**

Multimedia Networking 2-91

中科大高网-21秋

方法1: 最大限度利用尽力而为网络 (续)

- ❖ 带宽提供还是网络估算, 都需要估算端节点之间的多媒体应用性能需要
 - 网络端节点之间的 **通讯需求建模** (call级别, 分组级别) 负载随时间变化的情况
 - 良好定义的 **性能需求** (给定网络拓扑和带宽, 给出通信需求, 给出性能)
 - 以上2个因素代表者通信的需求
 - 根据预测端到端性能模型 (给定负载模型), 发现最小代价 **带宽分配** 从而满足所有用户需求的优化技术 (最少的部署代价满足所有用户的需要)?
 - 给定网络拓扑, 提供什么级别的带宽, 可以满足需求 (用户、分组或者连接级别的性能;)
 - 设计网络拓扑, 提供什么级别的带宽, 可以满足需求

Multimedia Networking 2-92

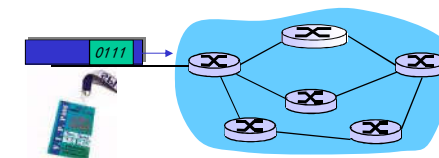
方法1：最大限度利用尽力而为网络（续）

- ❖ 互联网如果合理估算其网络部署，可以很好地支持多媒体应用达到合理的水准
- ❖ 应用情况：不尽如人意，不是所有ISP都部署足够的带宽
- ❖ 经济原因：用户愿意付足够多的费用给ISP（用于安装足够的带宽）吗？
 - 包月，ISP提供的服务很好，也多收不了费用
- ❖ 组织原因：端到端路径跨越多个ISP，需要他们合作提供很好的支持；

Multimedia Networking 2-93

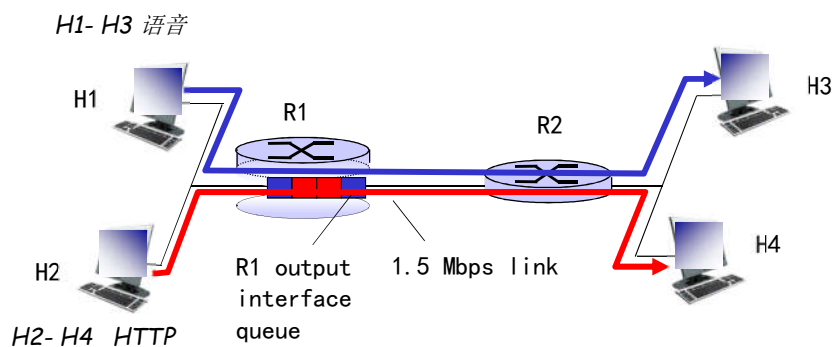
方法2：提供多种服务类型

- ❖ 到目前为止：最大限度利用尽力而为服务
 - one-size fits all service model
- ❖ 另外一种选项：多种服务类型
 - 将流量分成几种类型
 - 网络对待不同类型的流量提供不同服务（类比：VIP服务 vs 普通服务，头等舱）
- 分类的标准：根据类型，根据付费
- ❖ 粒度：对不同类型采用不同服务，在一个类型内部的**不同连接**不加区分：扩展性好
- ❖ 历史：ToS bits



Multimedia Networking 2-94

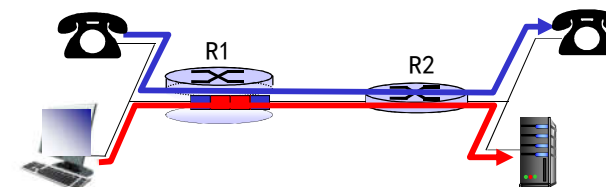
提供多种服务类型：场景



Multimedia Networking 2-95

场景1：混合HTTP和VoIP

- ❖ 例子：1Mbps VoIP, HTTP 共享1.5 Mbps link.
 - HTTP突发可能会拥塞路由器，导致语音分组丢失
 - 对策：http对延迟不敏感，希望将语音的优先级高于HTTP，http只是在缓冲有空的时候传



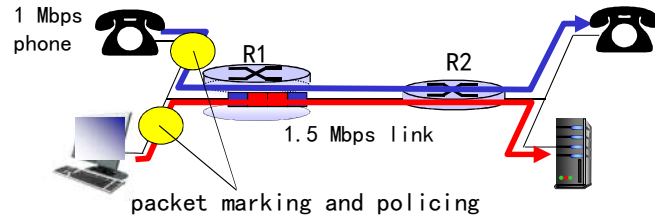
原则 1

需要将分组**标记**为不同类型，用于路由器加以区分；而且新增路由器的**服务策略**，可以根据不同类型采用相应策略

Multimedia Networking 2-96

提供多种服务类型

- ❖ 如果VoIP发送速率超过其声称的速率，http会饿死
 - 监管：强迫源端遵守其带宽分配
- ❖ 在网络边缘**标记、监管**，网络内部**调度**



原则 2

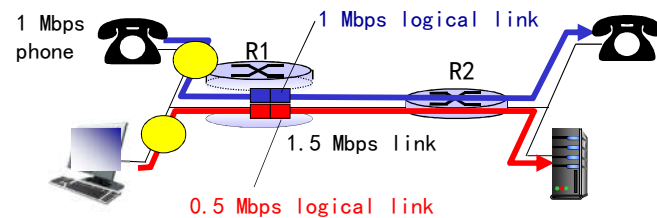
提供在类型间的保护（隔离），如果一类数据行为不当，也不会影响另外一类

提供多种服务类型

- ❖ 提供类间隔离的几种机制
 - 流量监管：
 - 向网络中注入分组的流量不会超过设定的条件；
 - 如果超过，丢弃或者延迟（整形），漏桶是最常用的监管机制，
 - 分类、标记和监管都在网络边缘主机或者边缘路由器上实现
 - 链路层分组调度机制：为每个类分配固定带宽，逻辑链路（带宽1.0Mbps或0.5Mbps）

提供多种服务类型

- ❖ 为流分配固定的（非共享）带宽：如果流没有使用完其分配给它的带宽，则对带宽的利用率较低

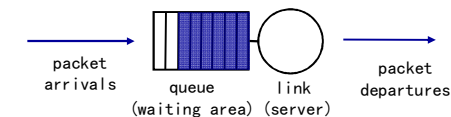


原则 3

在提供隔离的同时，与此同时最好尽最大可能有效利用资源

技术措施：调度和监管机制

- ❖ **调度**：选择下一个要通过链路传输的分组
- ❖ **FIFO (first in first out) scheduling**：按照分组到来的次序发送
 - 现实例子？
 - **丢弃策略**：如果分组到达一个满的队列，哪个分组将会被抛弃？
 - **tail drop**：丢弃刚到达的分组
 - **priority**：根据优先权丢失/移除分组
 - **random**：随机地丢弃/移除

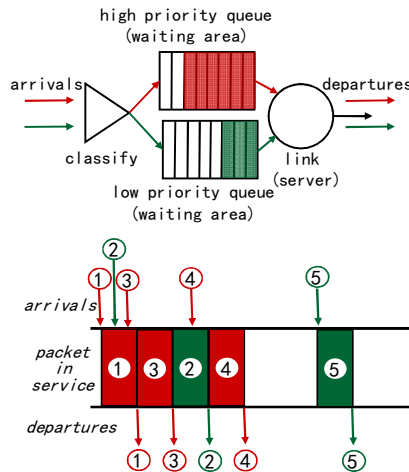


调度策略：优先权

优先权调度：发送最高优先权的分组

❖ 多类，不同类别有不同的优先权

- 类别可能依赖于标记或者其他头部字段，e.g. IP source/dest, port numbers, ds, etc.
- 先传高优先级的队列中的分组，除非没有
- 高（低）优先权中的分组传输次序：FIFO
- 现实生活中的例子？

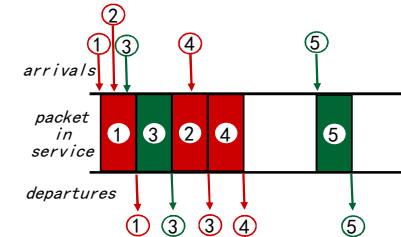


Multimedia Networking2-101

调度策略：其他的

Round Robin (RR) scheduling:

- ❖ 多类
- ❖ 循环扫描不同类型的队列，发送完一类的一个分组，再发送下一个类的一个分组，循环所有类
- ❖ 现实例子？

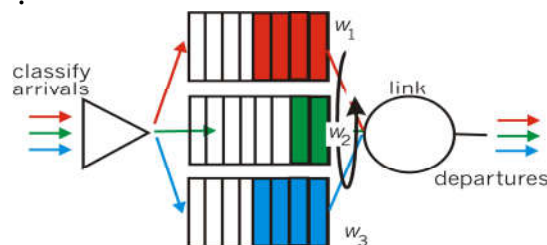


Multimedia Networking2-102

调度策略：其他的

Weighted Fair Queuing (WFQ):

- ❖ 一般化的Round Robin
- ❖ 在一段时间内，每个队列得到的服务时间是： $w_i / (\sum w_i) * t$ ，和权重成正比
- ❖ 每个类在每一个循环中获得不同权重的服务量
- ❖ 现实例子？



Multimedia Networking2-103

监管机制

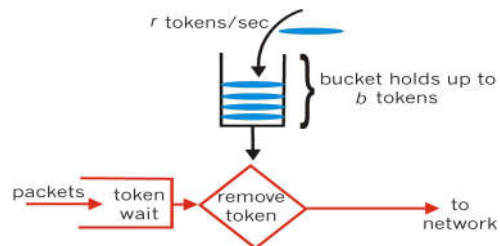
目标：限制流量不要超过声明的流量参数

- ❖ **平均速率average rate:** 一个相对长点的时间单元内允许发送多少个分组
 - ❖ 关键是：多少是一个合适的时间间隔：每秒100 分组或者6000个分组每分钟具有一样的平均值！
- ❖ **峰值速率peak rate:** 在一个相对短的时间内，能够往网络中注入分组的速率e.g., 1500 pps
- ❖ **最大突发长度burst size:** 在极短时间往网络中注入分组的速率，限制了连续不断向网络注入分组的数量（分组间没有间隔）

Multimedia Networking2-104

监管机制：实现问题

令牌桶token bucket: 限制向网络的输入的突发长度和平均速率



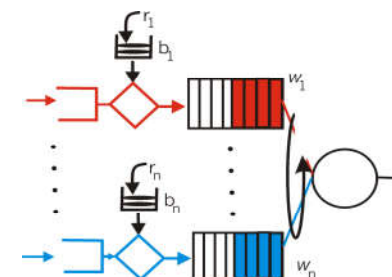
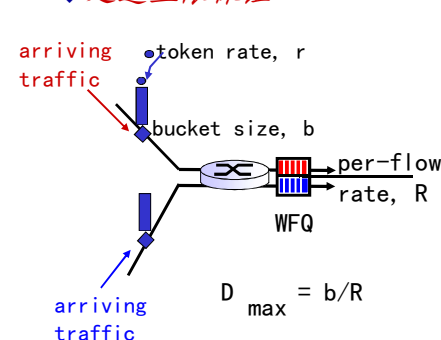
- ❖ 令牌桶可以容纳 **b** 个令牌
- ❖ 除非桶满，否则令牌每秒增加产生 **r** 个
- ❖ **t 时间内**：允许发送的分组数量不会超过 $(r \cdot t + b)$
- ❖ 限制了突发长度和平均速率

Multimedia Networking2-105

监管和QoS保证

❖ 令牌桶+WFQ 在一起可以提供延迟的上界保证
i. e., **QoS保证!**

- ❖ 每流带宽保证
- ❖ 延迟上限保证



Multimedia Networking2-106

区分服务：diffserv

- ❖ 前面介绍网络分类服务的原理，现在是真实协议
- ❖ 希望能够“区分（定性化）”服务类型
 - 例如：“表现像线路一样”
 - 相对服务上的区别：Platinum, Gold, Silver
- ❖ **可扩展性**：
 - 网络核心功能简单，根据标记对不同类型的分组提供不同的服务
 - 核心无需维护上百万流的状态，也无需信令建立流状态
 - 相对复杂的功能在网络边缘路由器（或者主机）上实现：分类，标记和监管
- ❖ 不定义具体的服务类型，仅提供建立服务类型的功能元件

Multimedia Networking2-107

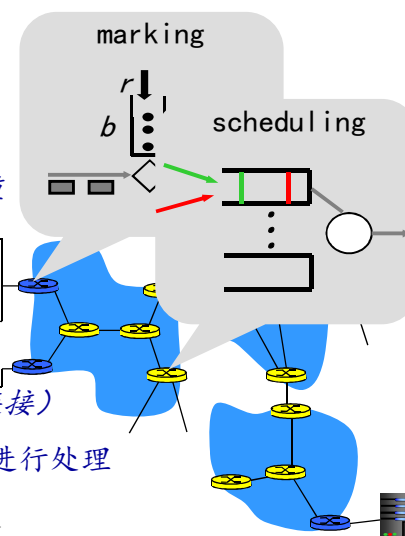
区分服务架构

边缘主机或者路由器：

- ❖ 建立流的状态，流管理
- ❖ 分组被标记不同类型，**DS**不同值
- ❖ 将分组标记为 **in-profile** 和 **out-profile**

核心路由器：

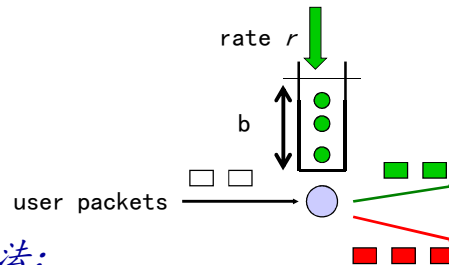
- ❖ 对每类流量进行管理（而不是连接）
- ❖ 根据每类分组关联的PHB对分组进行处理
- ❖ 基于边缘的标记进行缓冲和调度
- ❖ **in-profile**分组的优先权比**out-of-profile**分组要大些



Multimedia Networking2-108

边缘路由器的分组标记

- ❖ 概要：预先协商的速率 r ，桶的尺寸 b
- ❖ 基于针对每个流的概要，在边缘对分组进行标记



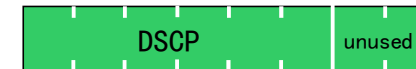
标记的可能用法：

- ❖ 基于类型的标记：不同类型分组标记为不同
- ❖ 类内的标记：遵守流协商的标记和违反流协定的标记不同
 - ❖ In-profile和out-profile

Multimedia Networking5-109

区分分组标记：细节

- ❖ 在IPv4中，分组在TOS字段进行标记；在IPv6中在Traffic Class中标记
- ❖ 采用6 bits来标记 Differentiated Service Code Point (DSCP)
 - 决定该分组将会接受的PHB
 - 2 bits当前不用



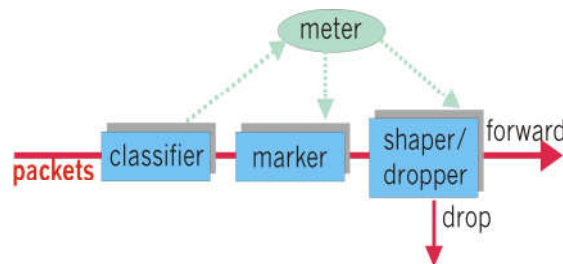
Multimedia Networking2-110

区分服务的关键功能1：边缘分类、标记和监管 (Classification, conditioning)

中科大高网-21秋

限制某类的流量注入速率：

- ❖ 测量用户的流量符合某个其同意的流量配置概要(e.g., rate, burst size)
- ❖ 如果不遵守协定，做整形



边缘路由器功能的逻辑视图

Multimedia Networking2-111

区分服务的关键功能2：Forwarding Per-hop Behavior (PHB)

中科大高网-21秋

- ❖ PHB：应用于diffser行为集的外部可观察行为
- ❖ 每类分组和PHB相关联，从而获得不同服务
- ❖ 没规定如何实现这种不同转发行为的具体机制
 - ❖ 只要能够满足外部可测量的性能指标，可以采用任何机制去实现，如：FCFS, WFQ等；
 - ❖ PHB是目的，而资源分配和实现机制是手段
- ❖ 不同PHB的性能差别应该可以被测量
- ❖ 例子：
 - 类型A在指定时间长度内，获得 $x\%$ 的输出链路带宽
 - 类型A分组要比类型B的分组先离开

Multimedia Networking2-112

转发PHB

2个已经被定义的PHB

- ❖ **显式转发***expedited forwarding*: 分组的转发速率等于或者超过特定速率
 - 拥有最小保证速率的逻辑链路
- ❖ **保证转发***assured forwarding*: 4 类流量
 - 将流量分成4个类型，每类被保证提供最小限度的缓存和带宽
 - 每个都有3个丢弃优先权分区

Multimedia Networking2-113

方法3：每流QoS保证

- ❖ 分组标记、监管、流量隔离、链路级调度
 - 一类服务要比一类性能要好，高优先权分组看不到低优先权分组
 - 合适的网络估算，高优先权的分组经历的延迟和丢包很少
- ❖ 使用上述机制，能够**保证**在会话期间的性能吗
 - 不行

Multimedia Networking2-115

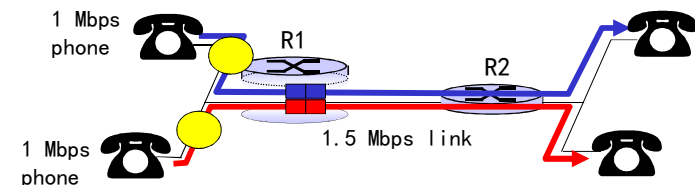
网络分组分类服务的观察

- ❖ 只在一个ISP内部实现区分服务，不足以保证端到端之间会话的性能
- ❖ 如果在网络中部署了区分服务，网络负载不（？）重情况下，区分服务通常比尽力而为服务好不到哪里去

Multimedia Networking2-114

方法3：每流QoS保证

- ❖ **常识**: 不能够支持超出链路容量的流量需求

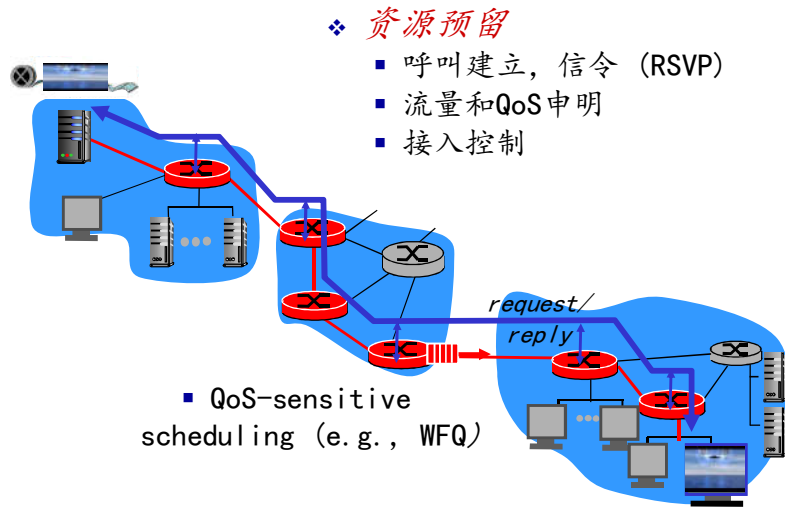


原则 4

呼叫准入: 流申明其需求，如果无法满足其需求网络可能阻塞该呼叫 (e. g., busy signal)

Multimedia Networking2-116

网络准入需要的一些协议和机制



Multimedia Networking2-117

总结

- ❖ 网络多媒体是当今最令人激动的应用, 趋势
- ❖ 网络多媒体应用分成3类
 - 存储流化, 实时交互, 实时非交互式
- ❖ 存储流化多媒体技术
 - UDP, HTTP, 自适应HTTP流化: 客户端缓存、预取、自适应媒体质量
 - CDN
- ❖ 实时交互式多媒体应用技术
 - 缓冲自适应播放延迟, 前向纠错, 丢失掩盖
- ❖ 3个协议: RTP, RTCP, SIP
- ❖ 网络机制
 - 最大限度利用尽力而为的网络服务, 区分服务, 资源预留

Multimedia Networking2-118