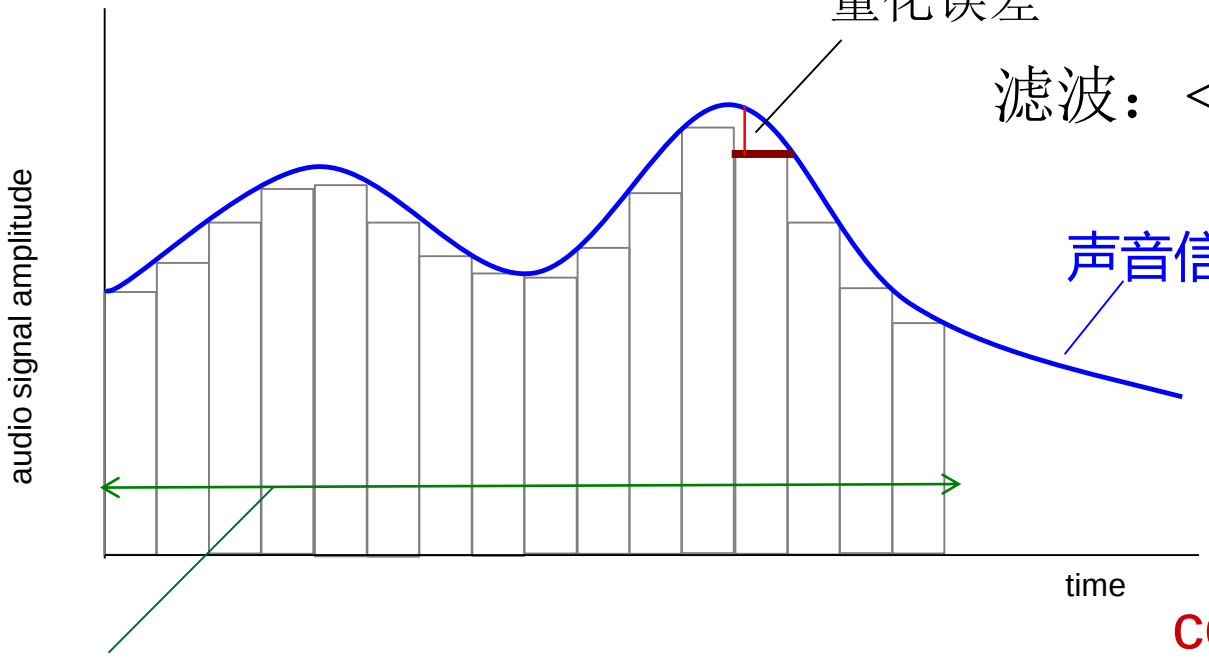
第八单元 多媒体网络

(Multimedia Networking)

内容

- 多媒体音频
- 多媒体视频
- 多媒体应用
- 流式存储视频
- IP语音
- RTP协议
- RTCP协议
- H.232
- 调度与排队策略
- 拥塞控制
- 拥塞预防
- 令牌桶算法
- 区分服务
- 综合服务
- MPLS

多媒体音频 (Multimedia: audio)



采样率

电话：8,000次/秒　每次8bit　5.3 kbps
CD：44,100次/秒　每次16bit　1.411 Mbps
MP3:　96, 128, 160 kbps

constant bit rate

variable bit rate

Niquist采样定理：采样频率大于信号最高频率的两倍，就可以完整地保留原始信号中的信息。

多媒体视频 (Multimedia: video)

frame i

frame i+1

- 一段每秒24帧的视频。每帧都是一幅图像。每幅图像可以用像素表示。每个像素可以用红绿蓝或者亮度色相饱和度两种方式表示。
- 可以对视频进行压缩。可以利用空间进行压缩 (图像内)，也可以利用时序进行压缩(相邻图像)。

多媒体应用

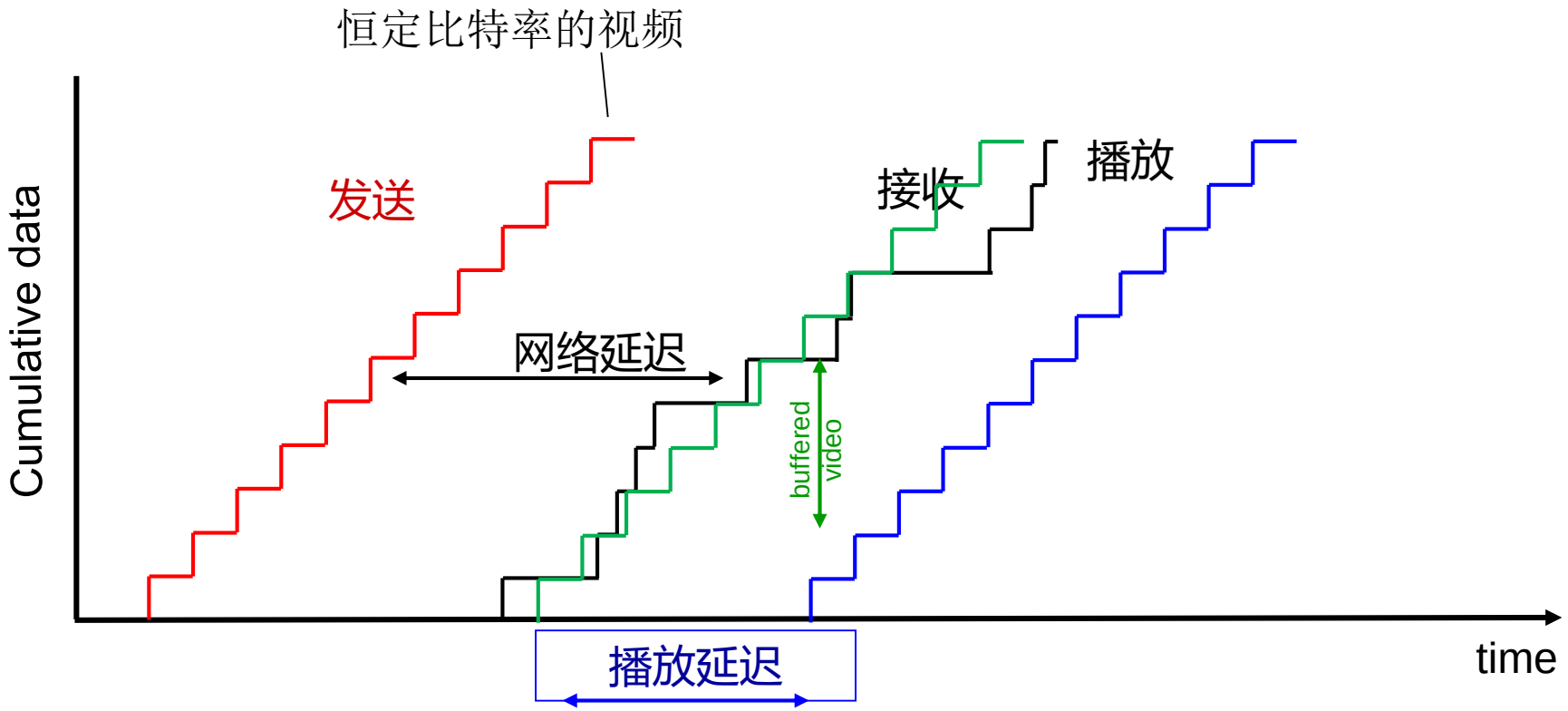
- 流式存储音频和视频
  - 流式(streaming): 可以在下载完整个文件之前播放
  - 存储(stored)：存放在服务器上，在客户端可以缓存
- 流式实况音频和视频
  - 例如，直播(live)体育比赛
- 会话式IP音频和视频（Skype、QQ、NetMeeting）
  - 人与人会话(conversation)的交互性是高度时延敏感的(delay-)。对于语音，150ms的时延不会被人察觉，150~400ms的时延可以接受，超过400ms的时延会使对话变得难以理解，令人沮丧。

* 音频和视频都是丢包容忍的(loss-tolerant)。偶然的丢失只会在音频或视频播放时出现干扰信号，而且可以采用技术处理进行掩盖。

流式存储视频

- 三种类型的系统
  - UDP流
    - RTP(Real-Time Transport Protocol)协议传送视频数据
    - RSTP(Real-Time Streaming Protocol)协议传送控制命令（暂停、重新开始、重定位）。媒体控制服务器执行RSTP。
  - HTTP流
    - 采用TCP连接。通过缓存消除TCP拥塞控制和重传机制带来的影响。
    - 可以穿越防火墙和NAT；容易部署，不需要媒体控制服务器。
  - 适应性HTTP流
    - 可以使用不同的编码速率进行播放（普通、高清、超高清）。

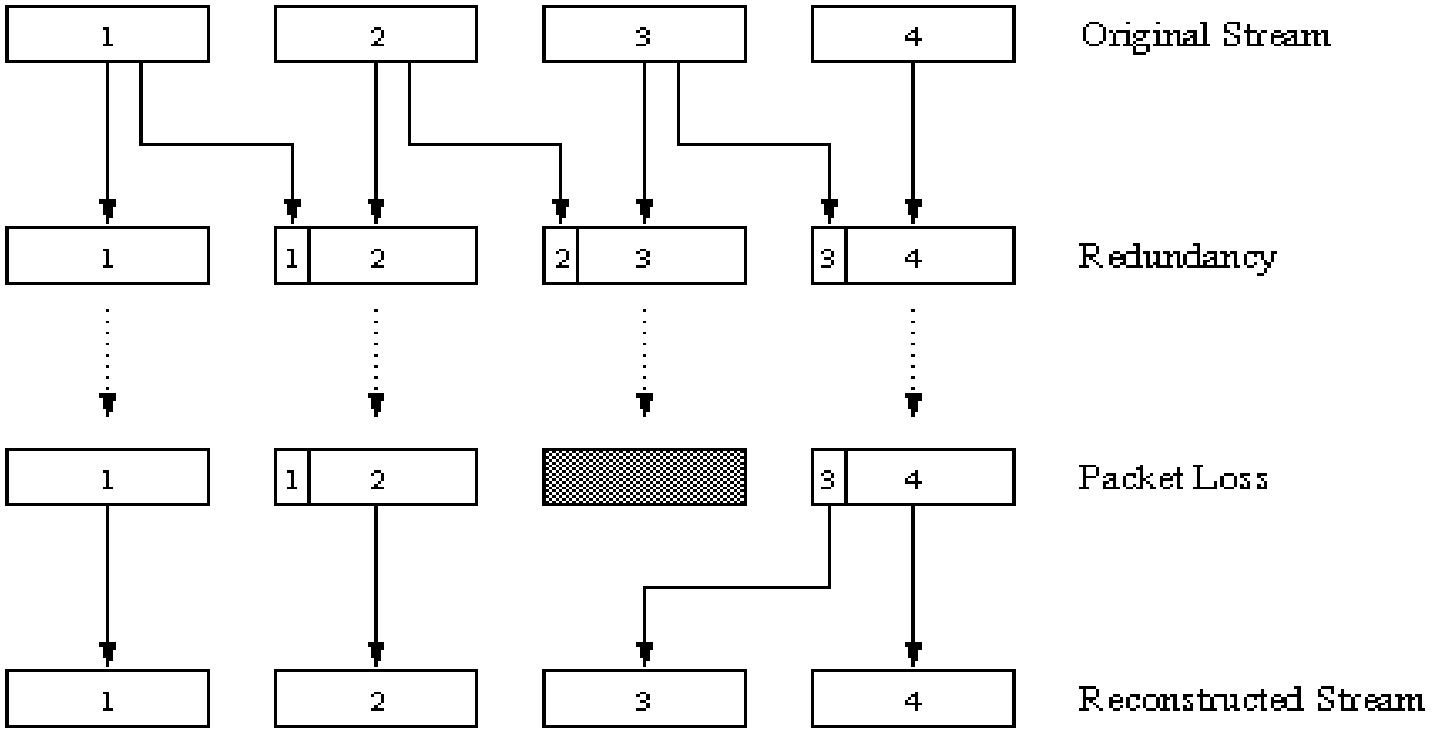
- 抖动
  - 客户端缓存和播放延迟：对网络延迟变长或抖动(jitter)进行补偿。



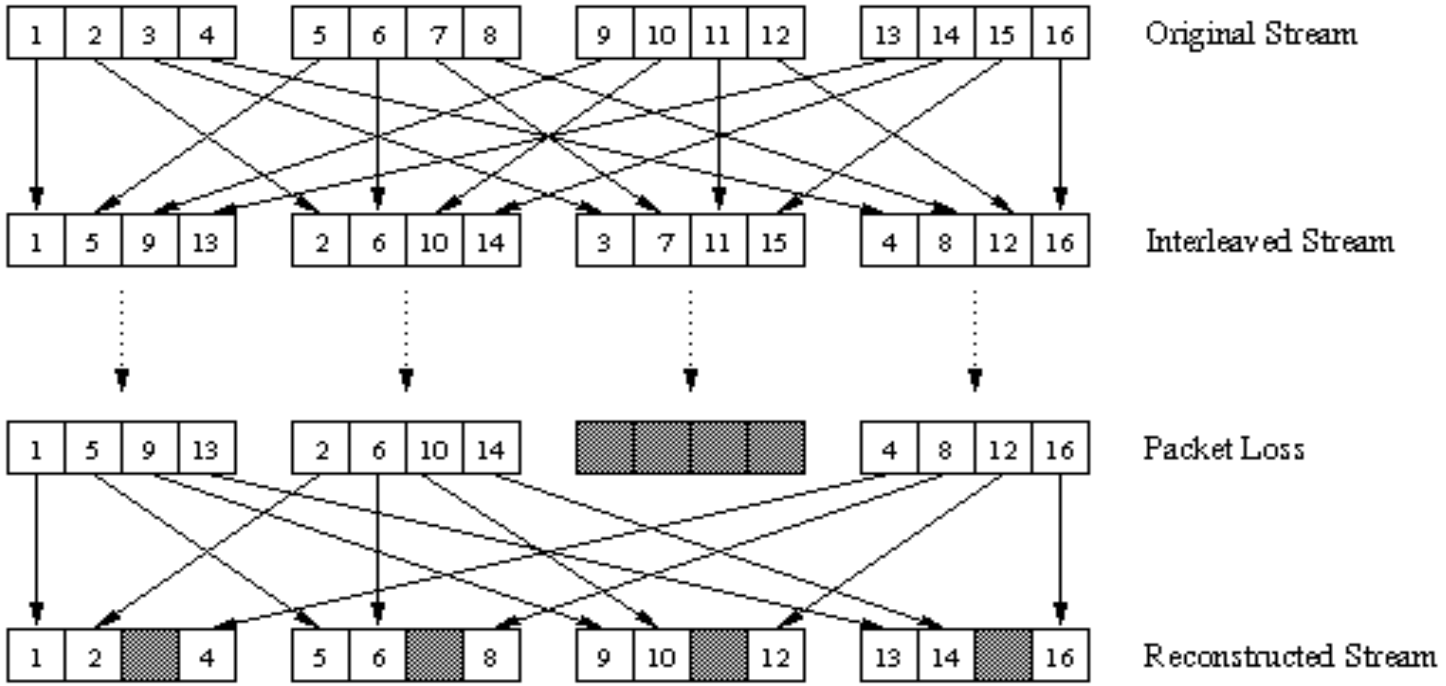
IP语音

- 采用UDP分组
  - 一个块中存放20ms产生的字节(20ms × 8000字节/秒=160字节)
  - 采用序号、时间戳和播放时延消除抖动
- 丢包恢复方案
  - 分组重复：用丢失之前的分组副本代替丢失的分组
  - 内插法：用丢失前后的分组副本内插形成一个合适的分组代替丢失的分组

- 捎带低质量的冗余信息


1
2
3
4
Original Stream
1
1
2
2
3
3
4
Redundancy
1
1
2
3
4
Packet Loss
1
2
3
4
Reconstructed Stream

- 发送交织音频

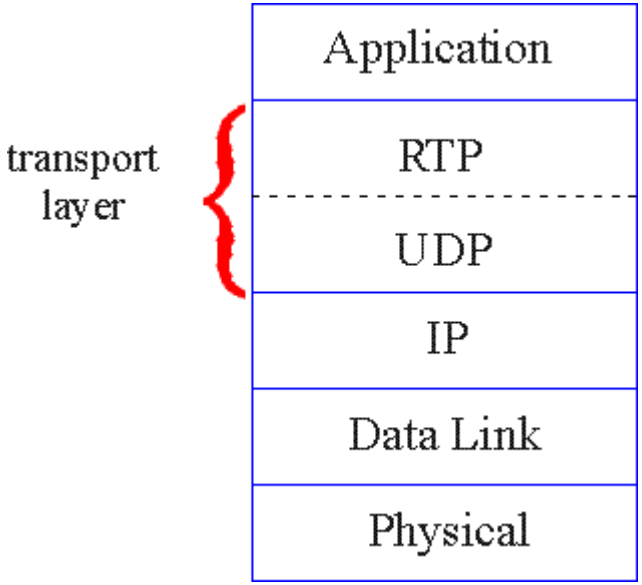

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Original Stream
1 5 9 13 2 6 10 14 3 7 11 15 4 8 12 16
Interleaved Stream
1 5 9 13 2 6 10 14 4 8 12 16
Packet Loss
1 2 4 5 6 8 9 10 12 13 14 16
Reconstructed Stream

RTP 协议 (Real-Time Protocol)

RFC 3550

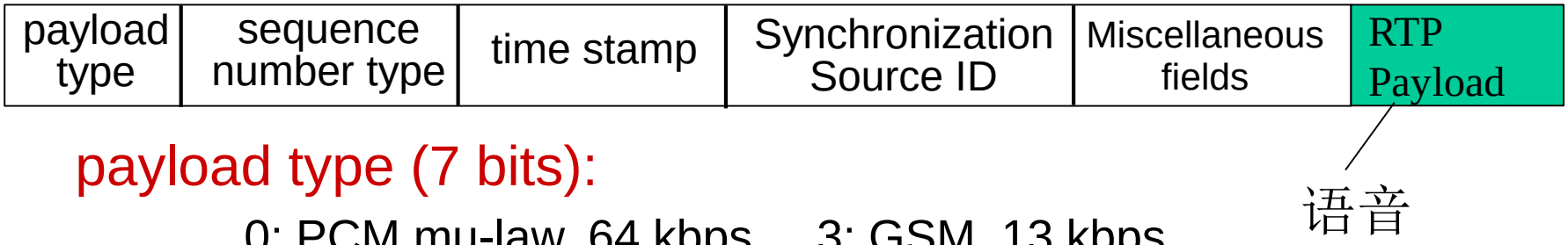
- RTP定义了传送音视频数据的包结构
- RTP包提供：
  - 有效载荷类型标识
  - 包序列号
  - 时间戳
- RTP运行在端系统上
- RTP包用UDP数据段封装
- 如果两个VoIP应用都采用RTP，它们将具有互操作性(interoperability)

- RTP运行在UDP之上



RTP例子

- 发送64 kbps PCM编码的语音



payload type (7 bits):

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

sequence # (16 bits): increment by one for each RTP packet sent. detect packet loss, restore packet sequence.

timestamp field (32 bits): sampling instant of first byte in this RTP data packet. Increase by about 20ms if sampling every 20ms.

SSRC field (32 bits): identifies source of RTP stream. Each stream in RTP session has distinct SSRC.

Payload field: audio chunk,160Bytes

RTCP协议 (Real-Time Control Protocol)

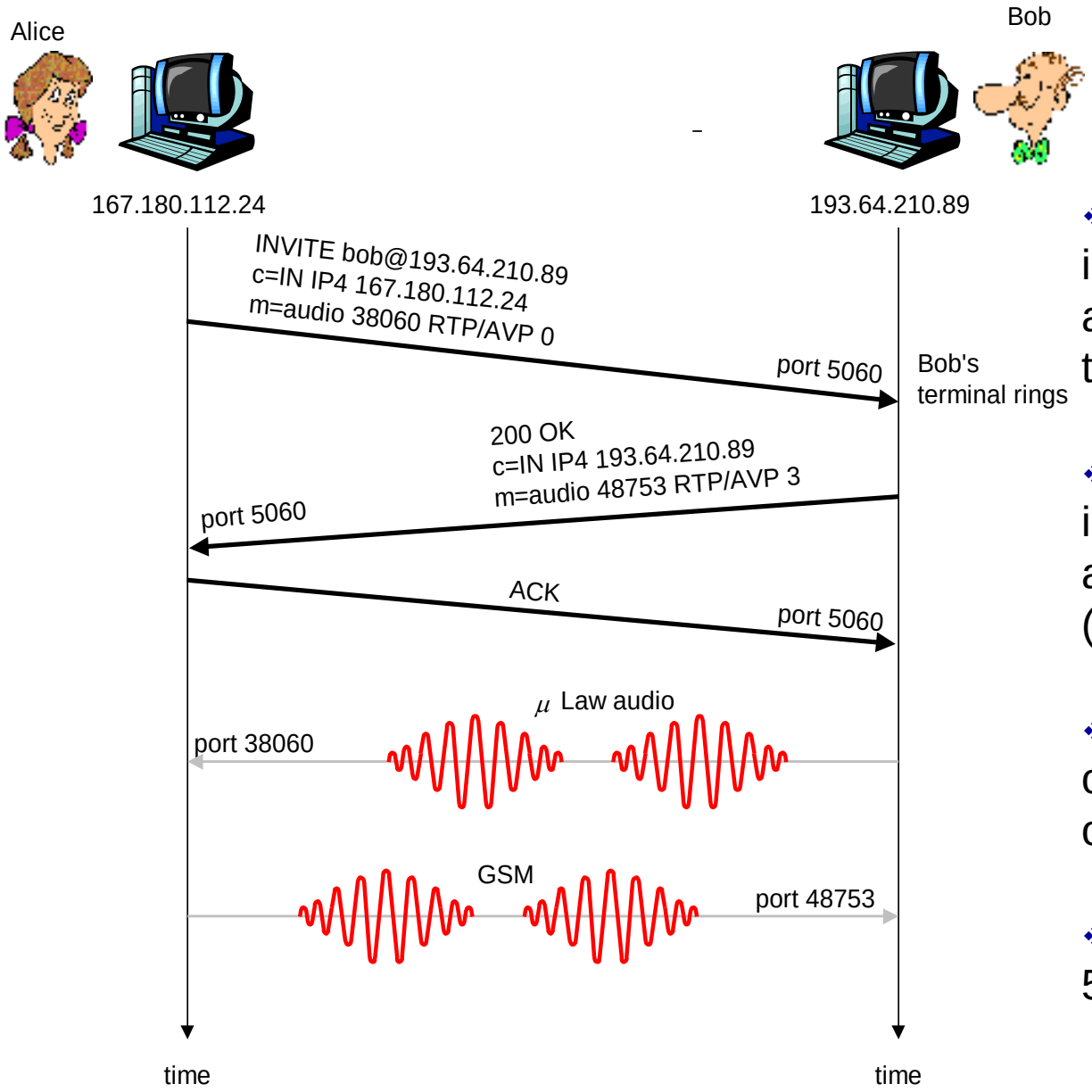
- 与RTP一起使用
- 每个参与者周期性发送RTCP包给其他参与者
- 每个RTCP可以包含两个报告
  - 发送者报告：发送的包数，发送的字节数，时间戳
  - 接收者报告：丢失的包数，最后的序号，平均间隔抖动
- 用于控制性能的反馈
  - 发送者可能基于反馈改变发送
- 当视频会议的人数增加时，RTCP可以通过多播把其占用的流量限制在会话带宽的5%以内。
- RTCP可以同步一个会话中的不同的媒体流，例如，一个视频流和一个音频流。

SIP协议 (Session Initiation Protocol)

[RFC 3261]

- SIP协议的远期目标
  - 所有电话呼叫，视频会议呼叫都可以通过因特网进行。
  - 可以用姓名或 e-mail地址而不是电话号码拨号
  - 无论被叫者漫游在哪里使用什么IP设备，都可以到达被叫者 (*只要被叫者希望这么做*)
- SIP协议提供的服务
  - SIP提供呼叫建立的机制:
    - 呼叫者要让被叫者知道他想建立一次通话
    - 呼叫者，被叫者可以在所有媒体流类型和编码达成一致
    - 可以结束通话
  - 确定被叫者IP地址:
    - 把助记符与当前IP地址进行映射
  - 呼叫管理:
    - 在通话期间可以增加新的媒体流，可以改变编码
    - 可以邀请其他人加入
    - 可以转移或保持通话

SIP举例: 用已知IP地址建立呼叫



- Alice's SIP invite message indicates her port number, IP address, encoding she prefers to receive (PCM μlaw)
- Bob's 200 OK message indicates his port number, IP address, preferred encoding (GSM)
- SIP messages can be sent over TCP or UDP; here sent over RTP/UDP
- default SIP port number is 5060

- 编解码协商:
  - 假设Bob不支持PCM μlaw编码器
  - Bob将用"606 Not Acceptable Reply", 并列出他的编码器。Alice从中选择一个并重新发送INVITE消息。
- 拒绝一个呼叫
  - Bob可以用 "忙," "离开," "需要付款," "禁用"等进行拒绝
- 媒体数据可以采用RTP或其他协议发送

SIP报文的例子

```
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
```

- 这里我们不知道Bob的IP 地址，但是知道SIP服务器的地址
- Alice发送和接收都是采用SIP的默认端口 5060。
- Alice在SIP报文头部说明了SIP客户端通过UDP发送和接收SIP报文

Notes:

- 采用HTTP报文格式
- 内容格式遵循sdp协议(session description protocol)
- Call-ID对任何呼叫都是唯一的

SIP注册机

- SIP服务器的一个功能就是注册机(registrar)。
- 当Bob启动SIP客户端, 客户端会把SIP注册报文发送到 Bob的注册服务器

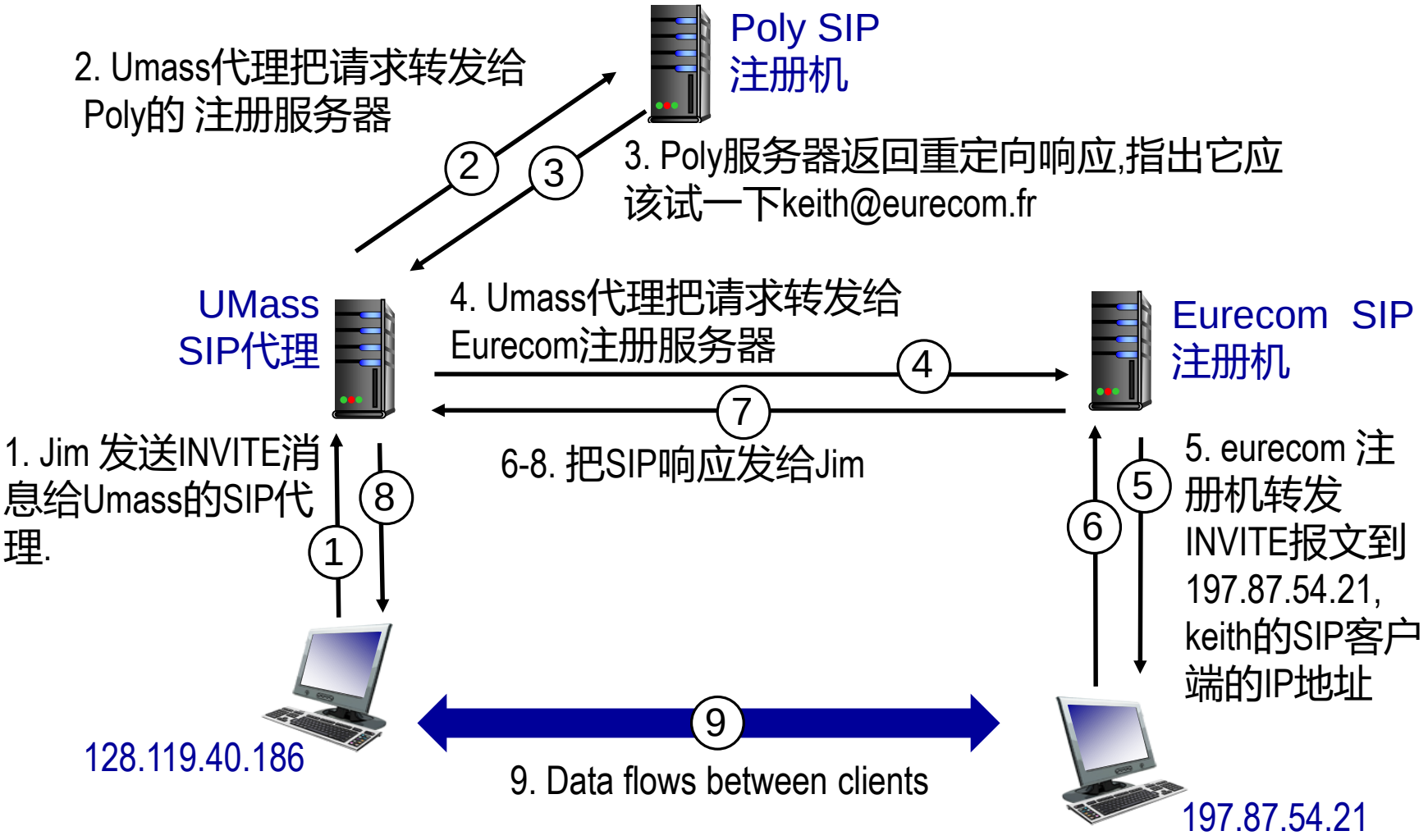
注册报文:

```
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代理(proxy)

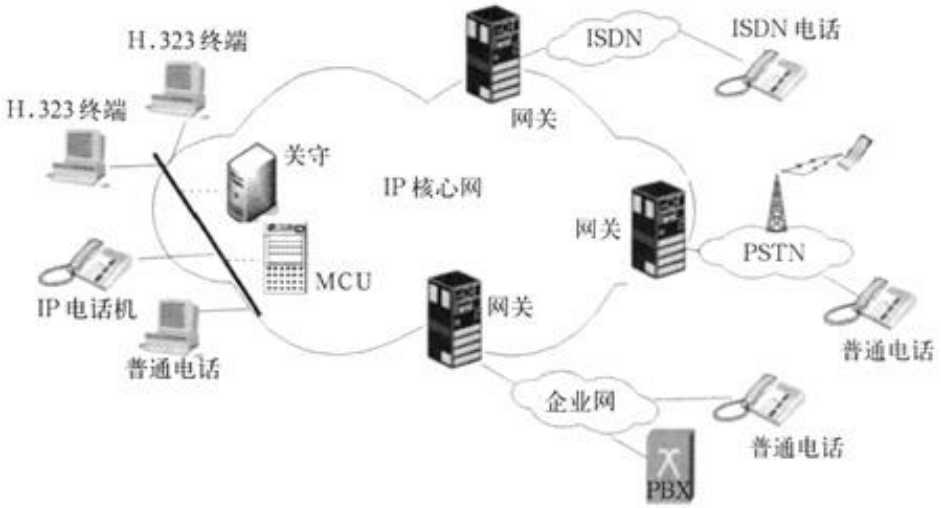
- SIP服务器的另一个功能: 代理(*proxy*)
- Alice发送邀请信息给她的代理服务器
  - 包含被叫者地址： sip:bob@domain.com
  - 代理负责把SIP消息选择路由传给被叫者，可能经过多个代理
- Bob也通过相同的代理集进行回应
- 代理把Bob的SIP响应发回给Alice
  - 包含Bob的 IP地址
- SIP代理与本地DNS服务器加TCP建立类似

SIP例子: jim@umass.edu 呼叫 keith@poly.edu



H.232

- H.323是另一个用于实时和交互多媒体的信令协议。
- H.323具有一个完整的用于多媒体会议的协议套件: 信令，注册，准入控制，传输，编解码。SIP只是单个组件，可以与RTP一起使用，也可以与其他协议和服务一起使用。
- H.323由ITU制定，很多概念来自电话，SIP由IETF制定，很多概念来自HTTP。SIP有Web风味， H.323 有电话风味。SIP采用KISS原则: Keep It Simple Stupid。



http://www.cww.net.cn/article/article.asp?id=91605&bid=2743

<table>
<tr><td colspan="2">音、视频应用</td><td colspan="4">终端控制与管理</td><td>数据应用</td></tr>
<tr><td>音频
G.711
G.722
G.723.1
G.728
G.729</td><td>视频
H.261
H.263
H.264</td><td rowspan="3">RTCP</td><td rowspan="3">H.225.0
RAS信令</td><td rowspan="3">H.225.0
呼叫信令</td><td rowspan="3">H.245
控制信令</td><td rowspan="2">T.124
T.125</td></tr>
<tr><td colspan="2">加密</td></tr>
<tr><td>RTP</td><td>认证</td><td rowspan="2">T.123</td></tr>
<tr><td colspan="4">不可靠传输（UDP）</td><td colspan="2">可靠传输（TCP）</td></tr>
<tr><td colspan="7">网络层（网络协议）</td></tr>
<tr><td colspan="7">链路层</td></tr>
<tr><td colspan="7">物理层</td></tr>
</table>

H.323

调度与排队策略

- FIFO队列(FIFO queuing)
  - 先到达接口队列的分组先从被调度离开接口。
- 优先权排队(priority queuing)
  - 优先权高的先发送。IOS使用high, medium, normal, low四个队列。只有优先权更高的队列为空时，才能发送低优先权队列的分组。问题：优先权低的队列可能被饿死。
- 公平排队(fair queuing)
  - 在输出端口每个流一个队列。按分组还是按字节发送？按字节计数更公平。

- 加权公平队列(weighted fair queuing,WFQ)
  - 根据服务要求分配发送速度。例如：视频服务或某种具有高优先权区分服务在每个滴答可以发送更多的字节。
  - 实际的做法是自动按流配置队列，每个流一个队列。每个流由源IP地址、目的IP地址、传输层协议(TCP或UDP)、源端口号、目的端口号五元组唯一定义。
  - 相同IPP(IP优先权，取值0~7)的流分配相同的带宽。更高IPP的流按比率获得更多的带宽。优先权为IPP=x的流获得的带宽是IPP=0的带宽的(x+1)倍。

- 自定义排队(custom queuing)
  - 按照自定义的分类方法最多形成16个队列。每个队列轮流获得发送权，并按照自定义的带宽百分比(字节计数)进行发送，默认带宽百分比相同。每个队列默认的排队分组最多为20个分组，采用队尾丢弃的方法(tail drop)。可见，公平队列只是自定义队列的一个特例。
- 基于类的WFQ(Class-based WFQ)和低延迟排队
  - CBWFQ为每个类定义一个队列，并为每个队列保留带宽。此时，WFQ只用于默认类的分组排队。CBWFQ最多提供64个类别，多个IP流可以属于一个类别。
  - 低延迟排队建立优先队列，只有优先队列都为空时才可以输出其它队列，可以用带宽限制其最大输出量，超过将被丢弃。

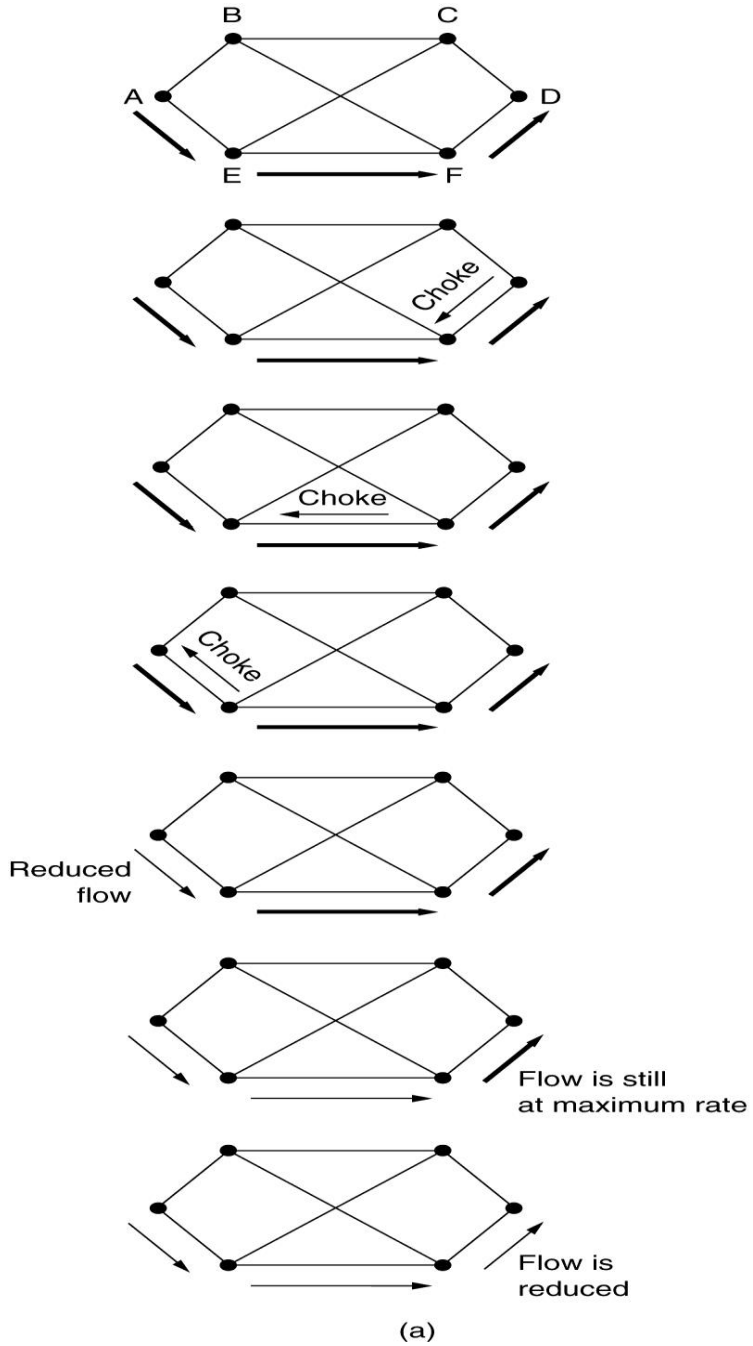
拥塞控制

- 抑制分组

  直接向源端发抑制分组，
  源端收到后降低数据发送速度。

- 逐跳抑制分组

  向源端发抑制分组，每个中间路由器收到抑制分组后都会降低数据发送速度。



(a) 抑制分组
(b) 逐跳抑制分组

拥塞预防

- 负载脱落(Load Shedding)

  通过丢弃分组解除拥塞状态（队列满）

  (1) 葡萄酒策略

  (2) 牛奶策略

  (3) 丢弃具有低优先权或高丢弃概率的分组。

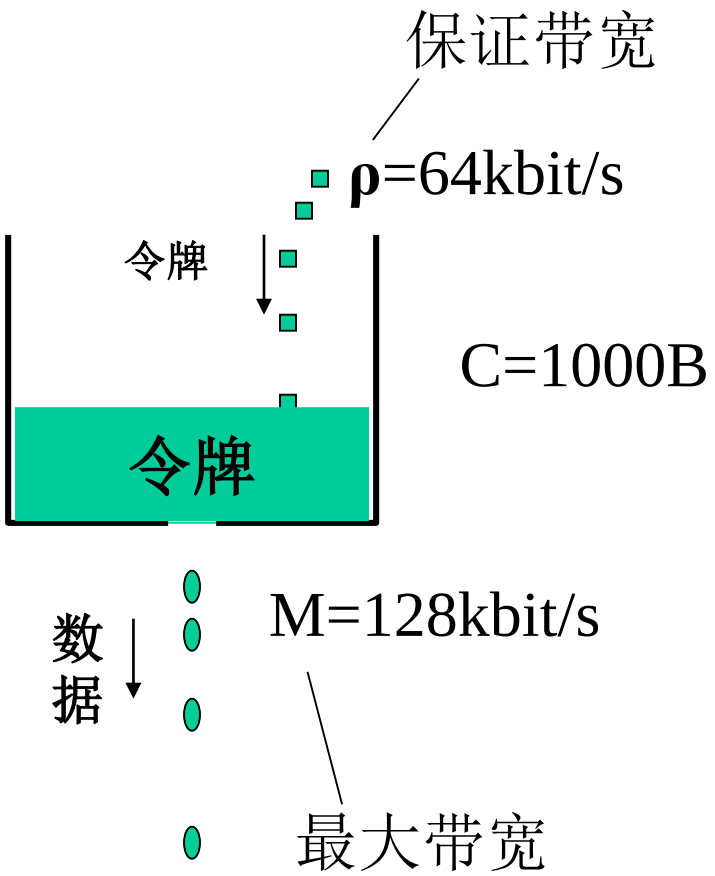
  帧中继把超出规定流量的部分标志为可丢弃

- 随机早期检测(RED，Random Early Detection)

  TCP的源端在确认数据超时到达(或收不到)时，将降低发送速率。当某个平均队列长度超过阈值时，向源端发送一个抑制分组或者随机选取分组丢弃而不报告源端。

令牌桶算法(Token Bucket)

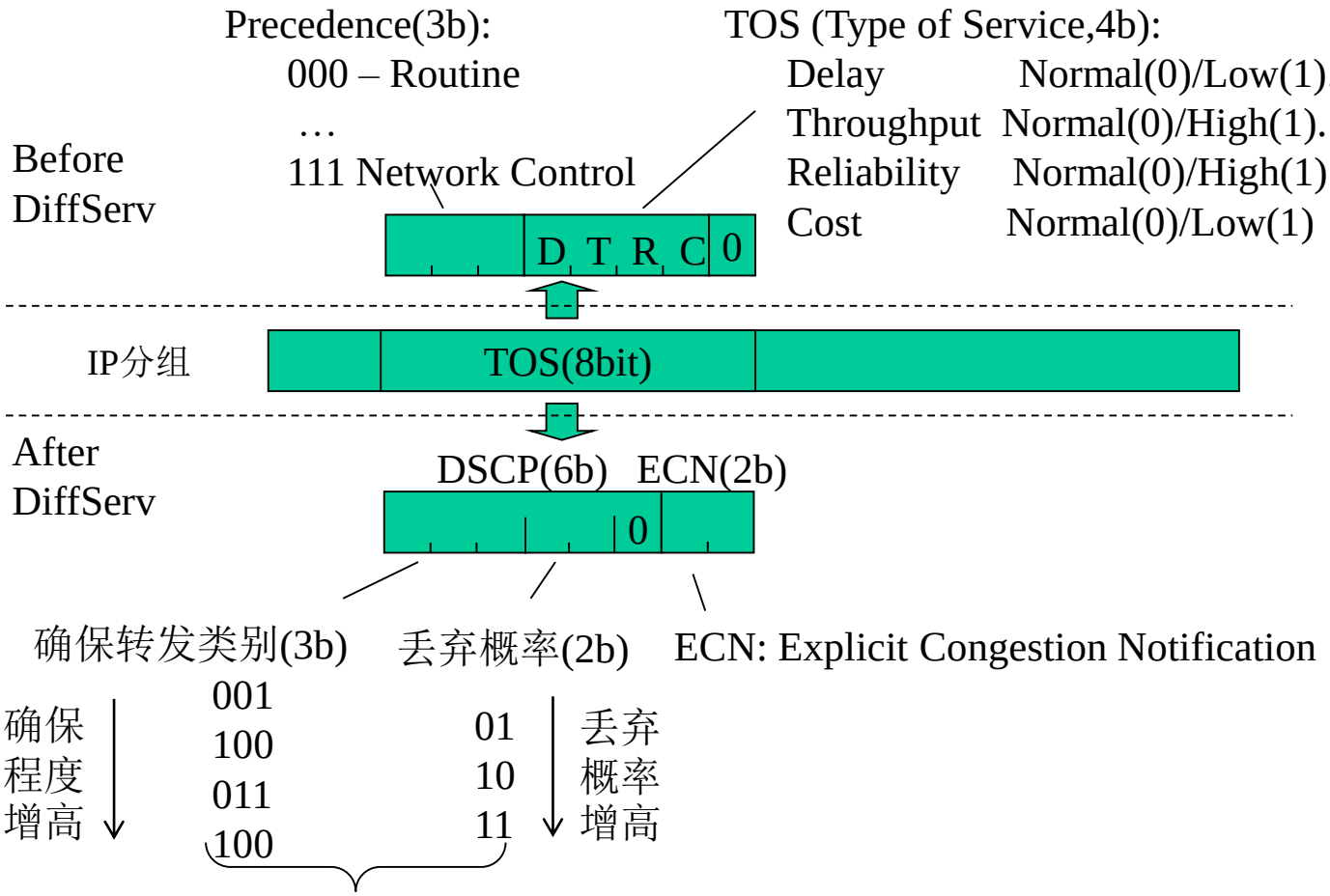
- 概述
  - 可以采用抑制接入速率的方法控制拥塞(漏桶)，但是对于大量突发数据的应用很不公平。这就出现了令牌桶算法。
- 令牌桶算法
  - 有多少令牌可以发送多少数据。
  - 令牌按照一定速率产生，并可以用令牌桶缓存起来。
  - 令牌桶具有一定的大小，缓存的令牌超过其桶的流量，则丢弃。



令牌产生速度为ρ，令牌桶C=1000B，最大数据速率是M=128kbit/s，突发时间长度S秒=?

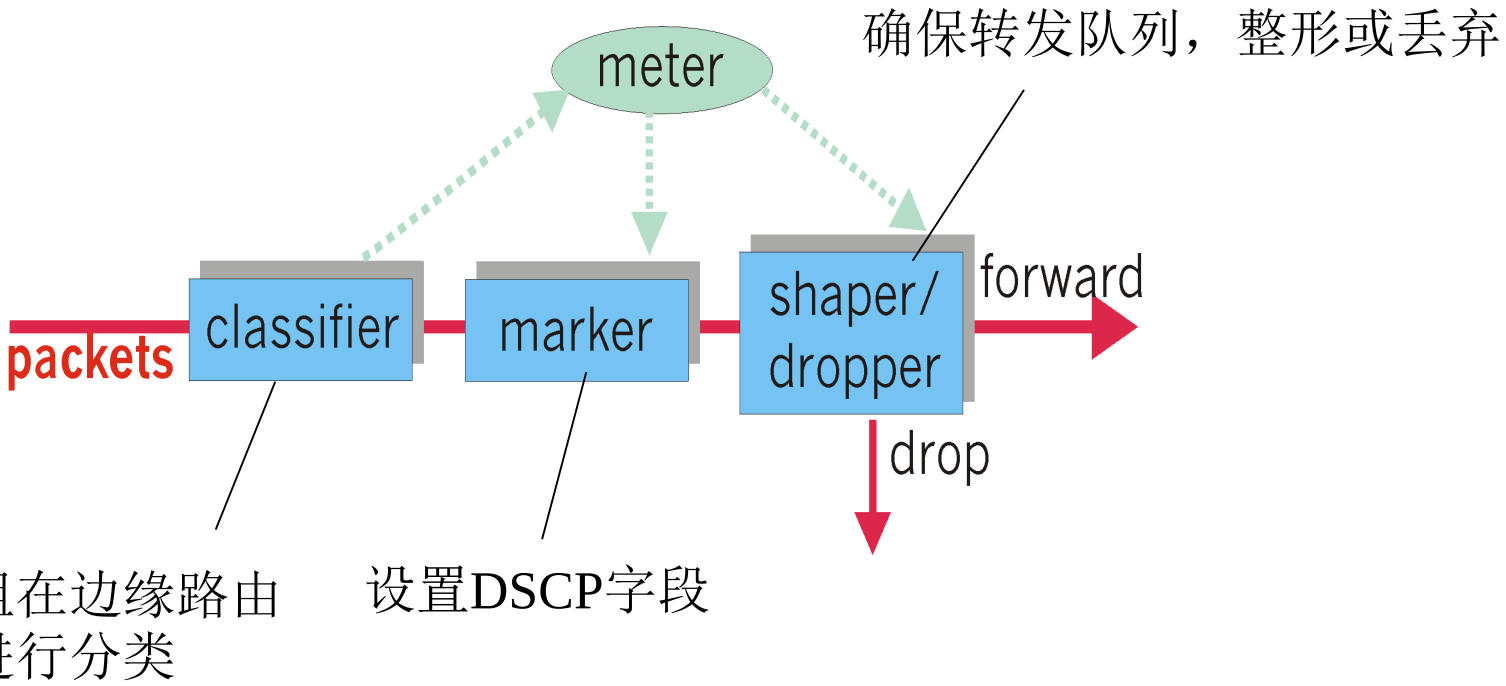
区分服务 (Differentiated Services, DiffServ)

- 概念



DSCP: 区分服务代码点(Differentiated Services Code Point)

- 每跳行为(Per-hop Behavior,PHB)

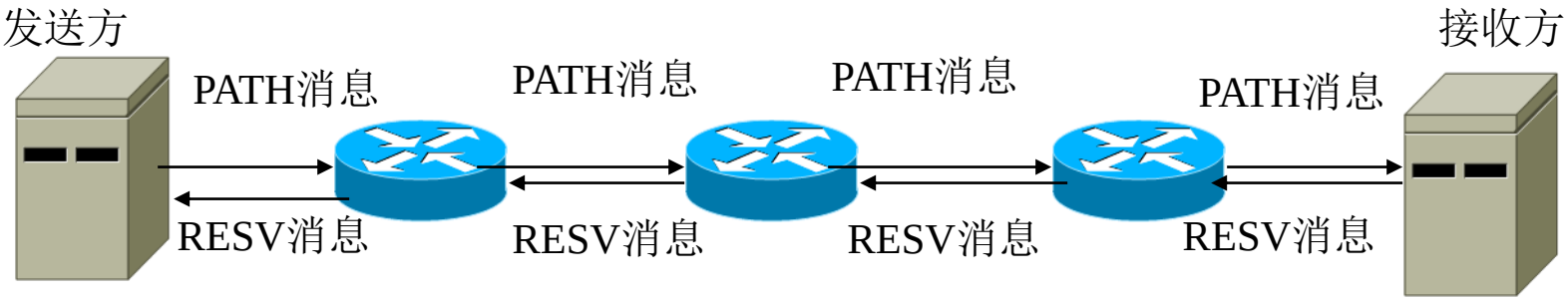


综合服务 (Integrated Services， IntServ)

- 综合服务为每一个流单独提供所需的服务。
  - 用流量说明 (TSpec)描述流的特性：发送速率，MTU。 用预留说明(RSpec)提出所要求的服务，例如，所需的带宽。
  - 当用TSpec向网络申请服务时，网络用准入控制(admission control)确定是否接受用TSpec申请的网络服务。
  - 当获得服务后，需要采用RSVP协议进行信令交互，以便预留所需的网络资源。
- 可申请服务类别：
  - 有保证服务(guaranteed service)提供有保证的带宽和延迟的服务。
  - 受控负载服务(controlled load service)提供允许丢失分组来保证所需的服务。该服务单独分配带宽，并通过控制负载总量保持合理的低负载。在网络重载时通过丢弃部分分组保证这些流的重要部分仍能被接收到。
  - 尽力服务(best-effort)：不提供任何类型的服务保证。

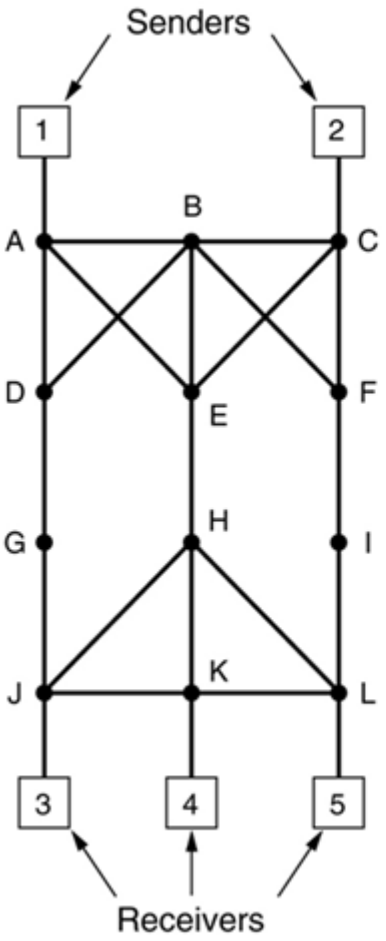
RSVP的工作过程：

- RSVP(Resource reSerVation Protocol)是一个每流(Per flow)协议，它为路径中的路径中的每个节点请求带宽预留。

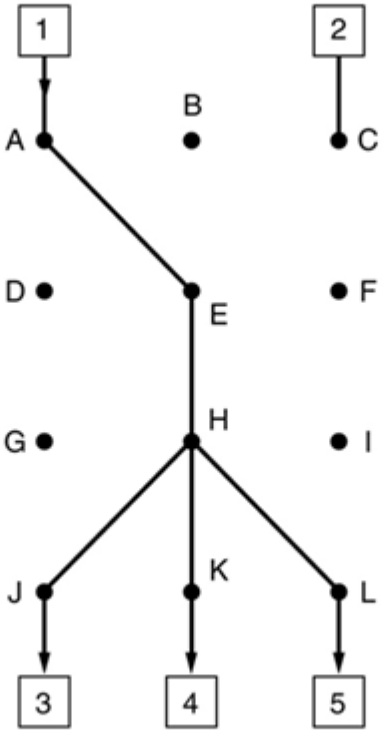


- 发送方每 30秒发送一次RSVP PATH 消息， 用来把TSpec发给接收方。TSpec 描述了发送方可能发送的流量特征。沿途路由器建立反向路由。接收到PATH分组后，接收方通过RESV消息沿反向路径把TSpec和RSpec给回发送方。 沿途路由器将按要求预留资源。如果不能预留资源，则通知接收方。
- 路径上的路由器保留流的软状态，并需要通过RESV周期性期刷新，否则，路由器将删除该软状态。 如果路径发生了变化，将自然转到新路径，原有路径上的路由器会删除软状态。

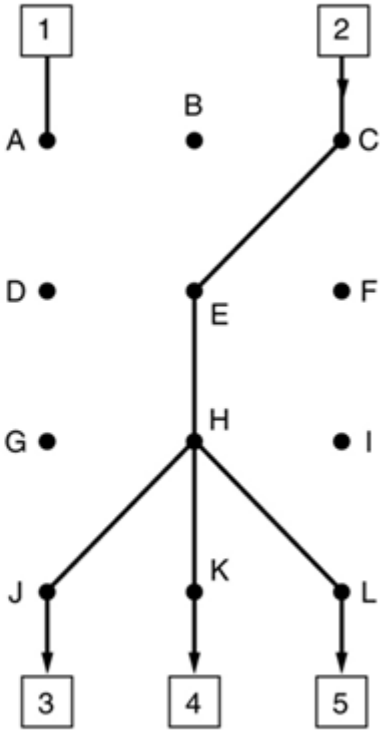
举例说明



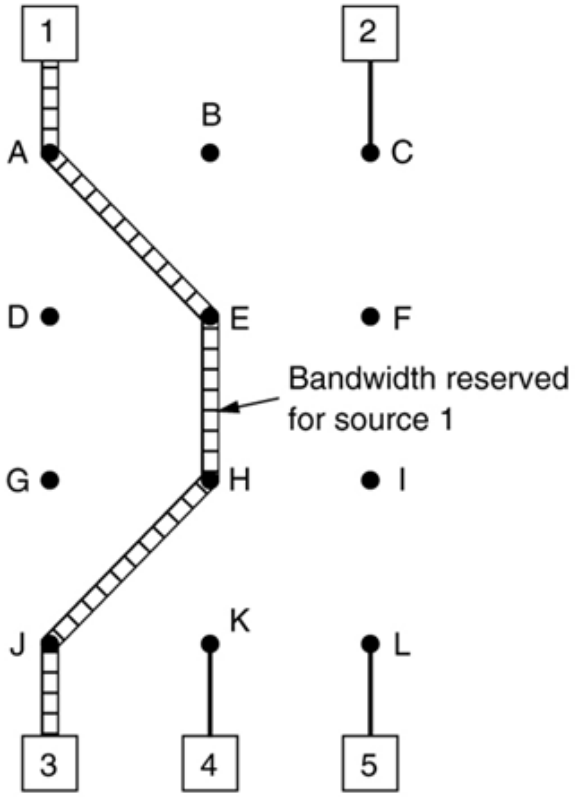
一个网络



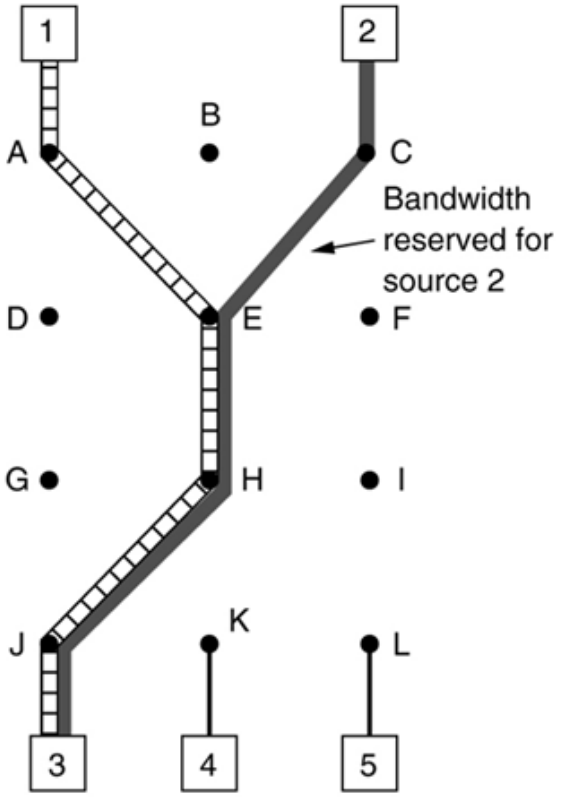
主机1的多播生成树



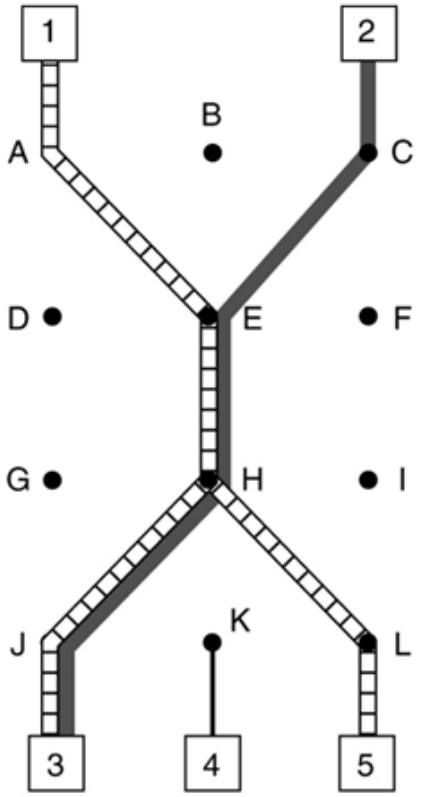
主机2的多播生成树



主机3向主机1请求
一个频道



主机3向主机2请求
第二个频道



主机5向主机1请求
一个频道

- InServ模型的优点
  - 概念简单,易于与网络策略管理集成。
  - 独立的每流QoS，非常适合语音呼叫。
  - 具有呼叫接入控制(CAC)能力，可以有效地防止拥塞。
- InServ模型的缺点：
  - 为每个流维护状态和交换信息将占用大量的带宽，因此，只适应小规模网络。
  - 周期性刷新需要可靠性保证。
  - 所有中间结点必须实现RSVP。

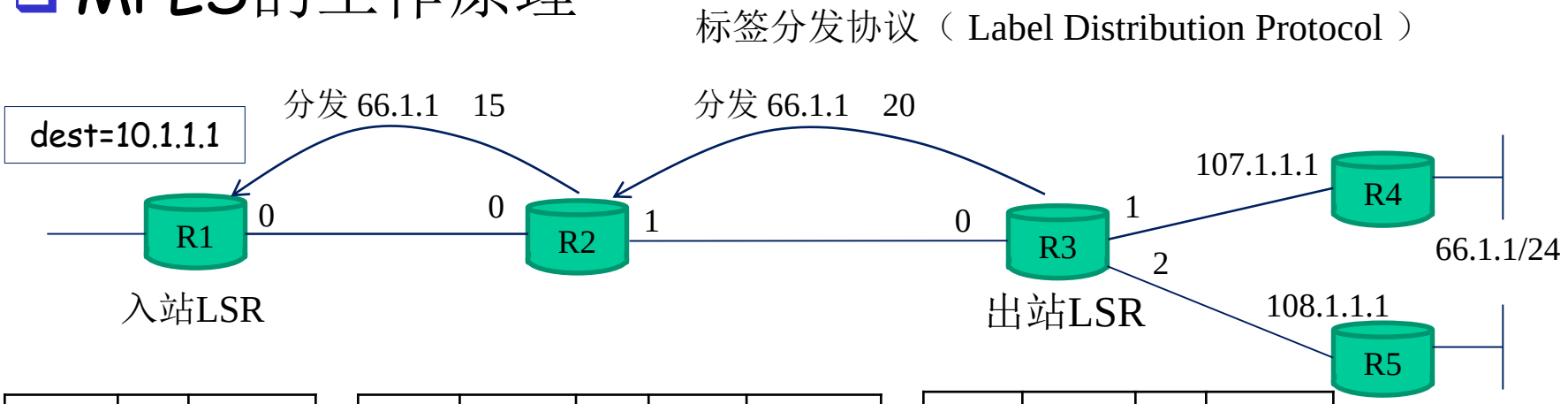
ATM

- ATM（Asynchronous Transfer Mode）出现在1980年代末和1990年代初。
- ATM是异步传输模式的简称。
- ATM是一种高速交换技术，它被设计在155Mbps或更高速度的链路上运行。
- ATM通过**在网络层提供有连接的服务**实现QoS(Quality of Service)。
- B-ISDN(Broadband-ISDN，综合服务数据网) 建立在ATM之上。
- 一个ATM信元(cell)53个字节:5B头部+48B数据

多协议标签交换

- MPLS的定义：
  - MPLS(Multi-Protocol Labeling Switching)通过在帧头和其有效载荷(上层协议)之间加入标签(一种VCI)，在因特网中实现标签转发 (类似虚电路）。由于该方法可用于各种协议的转发，所以也称为多协议标签转发。
- MPLS的应用：
  - BGP MPLS：使不具备BGP转发能力的内部路由器可以利用隧道转发AS的中转流量。
  - MPLS VPN：利用ISP提供的"虚电路"建立VPN。

MPLS的工作原理



prefix	int	remote label
66.1.1	0	
10.1.1	0	
…	…	…

label	prefix	int	next hop	remote label
15	66.1.1	1	R3	
16	10.1.1	1	R3	
…	…	…		…

label	prefix	int	Next Hop
20	66.1.1	1	R4
21	10.1.1	2	R5
…	…	…	…

LFIB：Label Forwarding Infomation Base

FEC：*forwarding equivalence class. a set of packets that* are to receive the same forwarding treatment in a particular router.

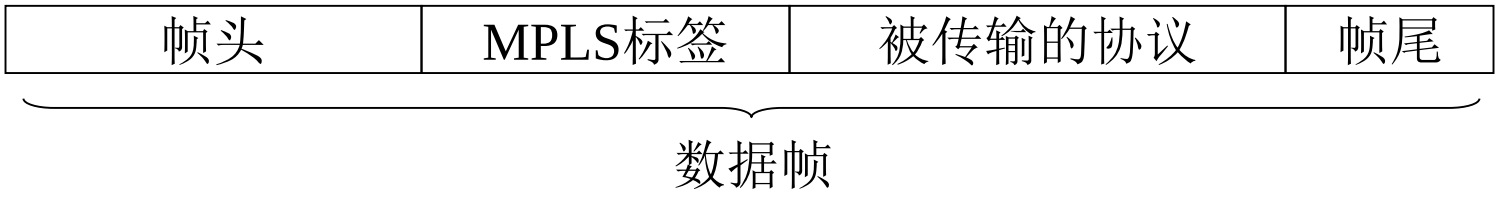
标签取值：基于接口；基于LSR(Label Switching Router).

本地标签也称为入站标签
远程标签也称为出站标签

标签分发协议(Label Distribution Protocol)

1. 路由器通过路由协议产生路由表。
2. 运行MPLS的路由器为路由表中的路由分配标签，作为该路由的本地标签。
3. 通过标签分发协议发现其MPLS邻居，并将<路由,标签>通告给其MPLS邻居。
4. 路由器将其下一跳路由器通告的标签加到其转发表中，作为该路由的远程标签。

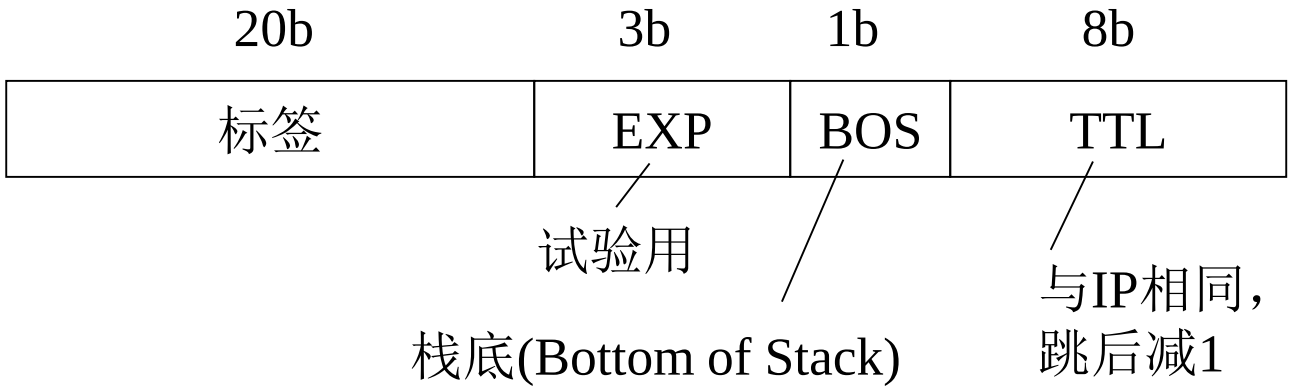
带标签的报文封装



帧头部的字段指明后面是MPLS标签：

帧	字段名	字段值
PPP	类型	0x0281
以太网802.3 LLC/SNAP	协议	0x8847
HDLC(CISCO)	协议	0x8847
帧中继	NLPID	0x80

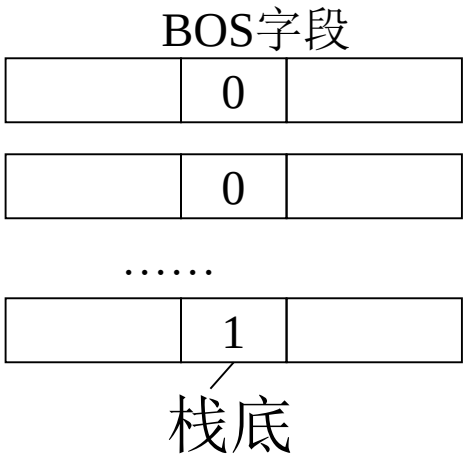
标签的格式



EXP: 用于分组排队和丢弃算法；

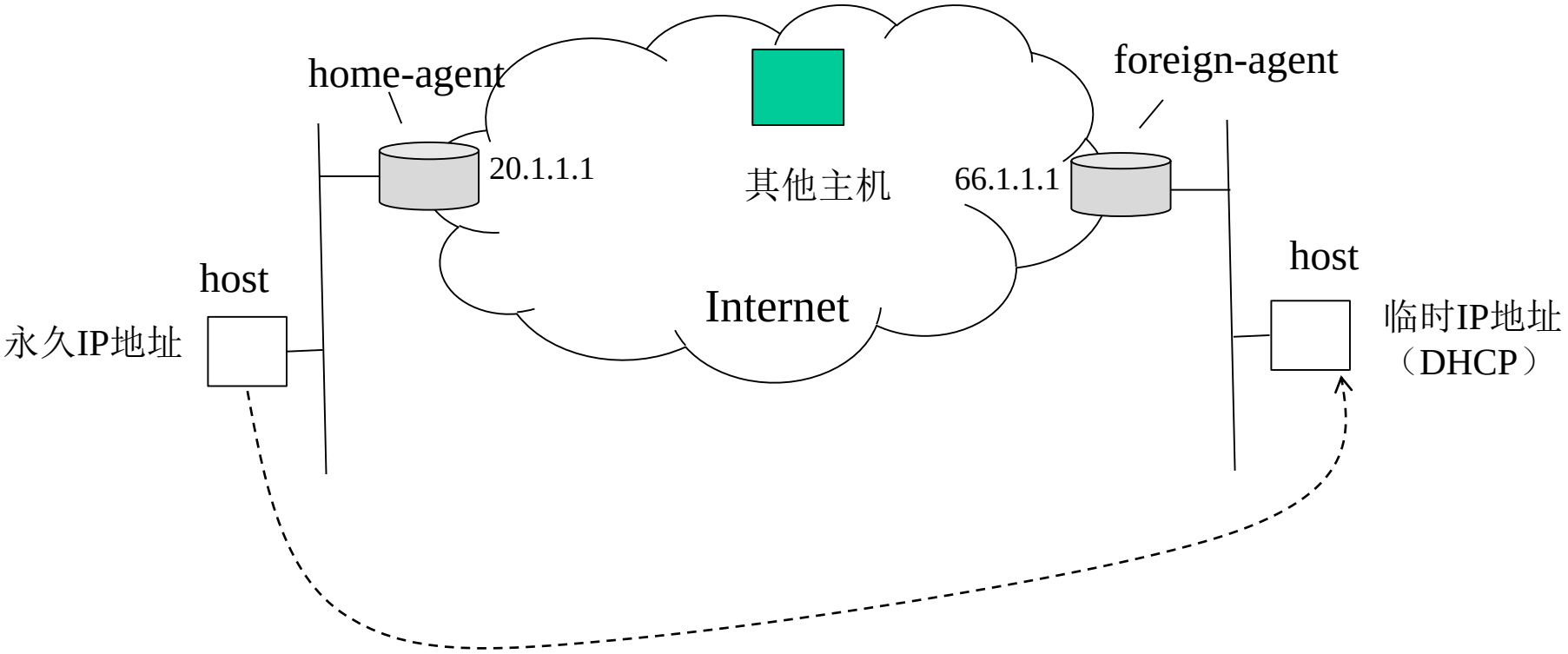
TTL: 提供传统的IP TTL功能。加标签时用IP TTL的值填入，去除标签时把该值填入IP分组的TTL。

多标签



移动IP

如果一台主机从一个网络移至另一个网络时希望其他主机依然可以使用其永久IP预期通信，如何设计协议？



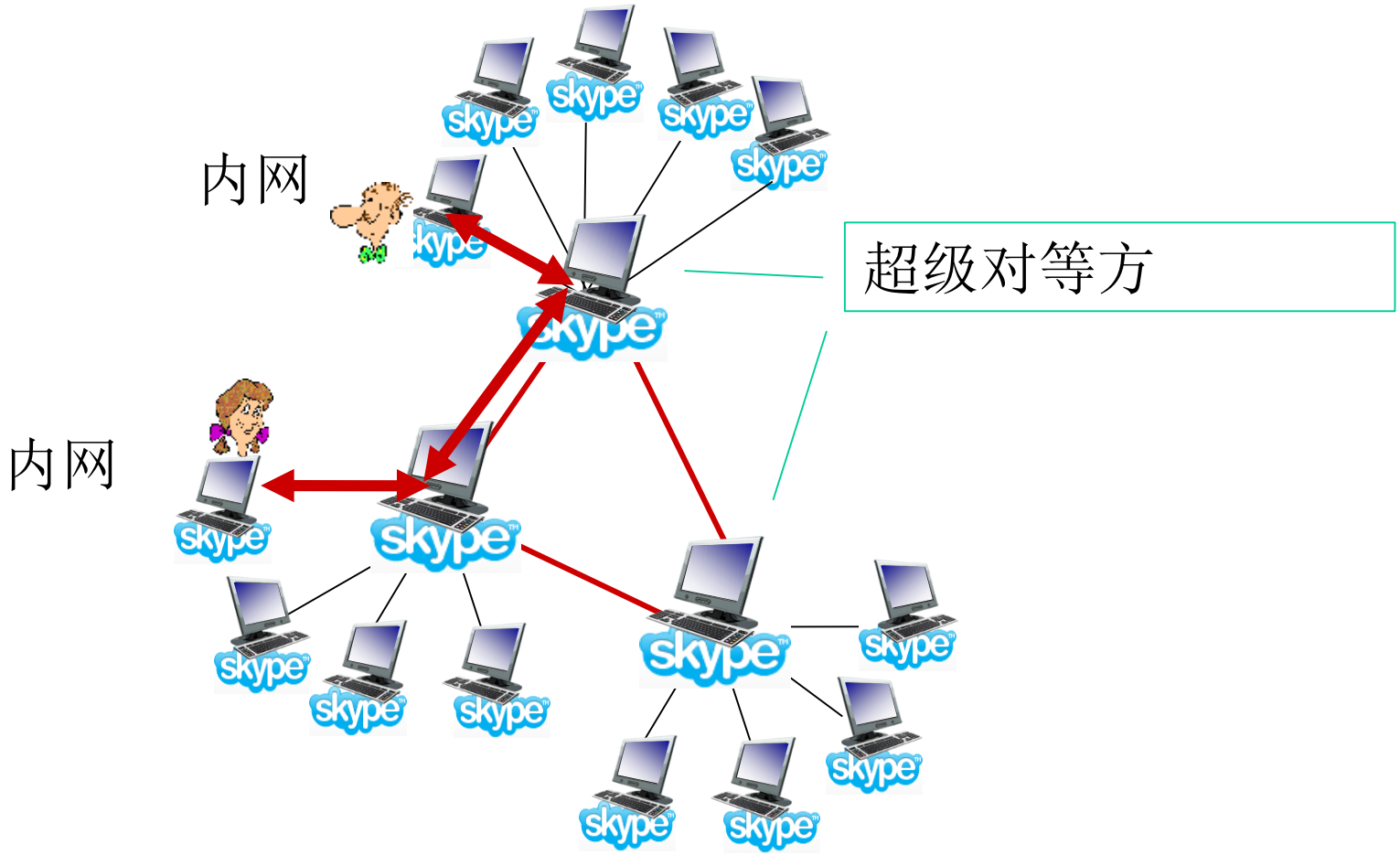
可以与其它主机直接通信吗？安全吗？

内容分发

- 内容分发网
  - 内容分发网(Content Distribution Network，CDN)管理存放了视频副本的放在不同位置的服务器，每个用户的请求将被定向到具有最好用户体验的服务器。

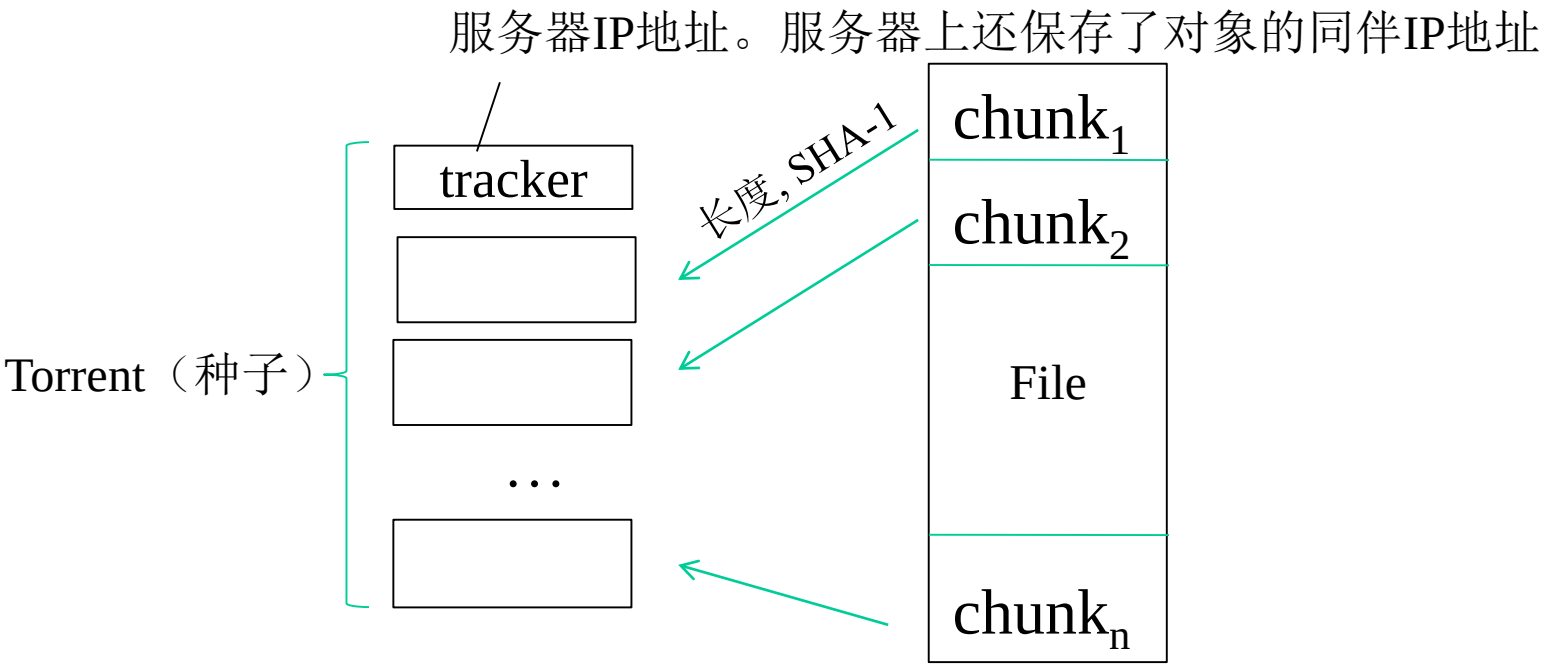
采用P2P技术

内网用户和内网用户之间要通过超级对等方进行转发，其它情况可以直接传送。



- P2P覆盖网络（overlay network）的优点
  - 因为CDN安装很多服务器，成本很高，如果可以利用用户存储文件，然后让有需要者用这些用户处下载，成本则会大大降低。在参与人数很多的情况下，P2P的下载速率可以远大于CDN。很多视频公司都采用CDN和P2P结合的方式。
- P2P覆盖网络的类型
  - 有**集中式网络**和**非集中式网络**。集中式网络的中心节点保存了文件存放点的信息。非集中式网络没有中心节点。
  - 非集中式网络可以通过**非结构化方法**或**结构化方法**保存文件存放地点的信息。**非结构化方法**随意存放和关联，用扩散方式查找保存文件存放地点的信息；**结构化方法**采用索引关联，利用Hash函数查找保存文件存放地点的信息。

BitTorrent



- 通过种子可以找到服务器(server)，服务器上保存了相同文件的服务器(server)及其伙伴(peers)形成的群(swarm)，伙伴称为播种者(seeders)。
- 一个加入群的伙伴可以同时从多个伙伴处下载不同的块。较少份数的块将首先被下载。正在下载的伙伴也把自己的块(chunk)提供给其它伙伴下载。
- 只下载不提供上传者将被噎住(choked)。正在下载而不做贡献的称为搭便车者(free-rider)或吸血者(leecher)。

- Gnutella
  - 采用非结构化的方法，每个节点连接已知的节点，并用扩散的方法查询对象(例如，歌曲)，用TTL和标识防止不断绕圈。
- Chord
  - 采用分布散列表(DHT)方法：用节点的身份信息(例如：IP地址)和对象的内容(例如：所有字节)都通过散列函数映射到同一个ID空间，分别称为节点ID和对象ID。由于这个ID空间很大，这个空间的很多ID并没有节点ID对应。
  - 如果一个节点拥有一个对象，它就把一个对象指针(该节点的IP地址）保存在这个对象ID的后继节点。每个节点保存一张索引表。利用索引表可以从任何节点开始用很快的速度找到一个对象ID的后继节点，然后就可以得到该对象的对象指针，最后通过该指针取到该对象。
  - 节点加入时要找到其前驱节点和后继节点，并修改它们的索引表，还要从后继节点中拷贝部分对象指针（ID在该节点之前的对象）。
  - 节点离开时也要维护节点ID的前驱和后继节点的索引表，还要把自己的对象指针拷贝到后继节点中。
  - 每个节点保存的对象指针也要在其后的后继节点中保存多份，以免节点失败时丢失。