

山东省精品课程

计算机网络

主讲人：李登道



本章的主要内容:

- 8.1 概述
- 8.2 流式存储音频/视频
- 8.3 交互式音频/视频
- 8.4 改进“尽最大努力交付”的服务



8.1 概述

计算机网络最初是为传送数据信息设计的。因特网IP层提供的“尽最大努力交付”服务，以及每一个分组独立交付的策略，对传送数据信息也是很合适的。

因特网使用的 TCP 协议可以很好地解决网络不能提供可靠交付这一问题。

多媒体信息的特点：

多媒体信息（包括声音和图像信息）与非多媒体信息有很大的区别。多媒体信息的信息量往往很大。

在传输多媒体数据时，对时延和时延抖动均有较高的要求。多媒体数据往往是实时数据，实时数据通常指的是：在发送这些数据的同时，接收端边接收边播放。

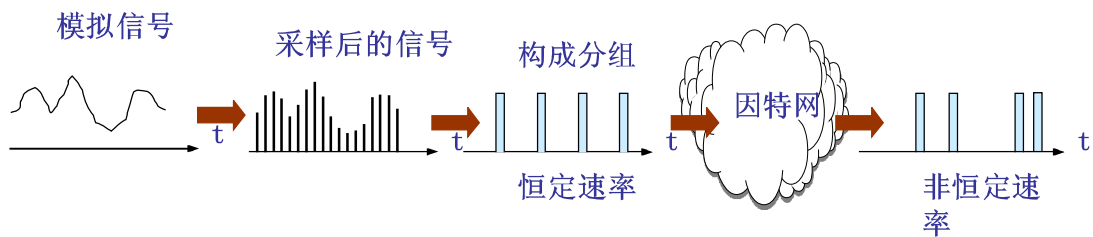


8.1 概述

因特网是非等时的

模拟的多媒体信号经过采样和模数转换变为数字信号，再组装成分组。这些分组的发送速率是**恒定的（等时的）**。

传统的因特网本身是**非等时的**。因此经过因特网的分组变成了**非恒定速率**的分组。

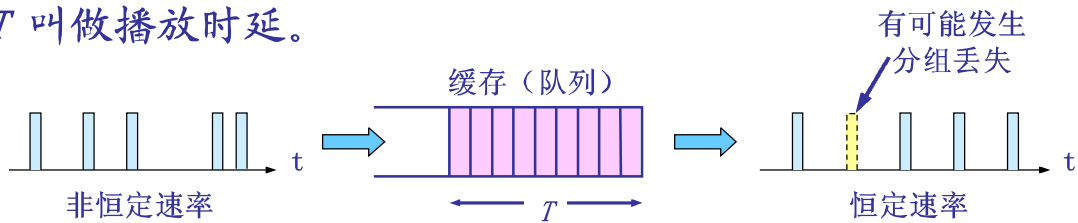




8.1 概述

在接收端设置缓存

接收端需设置适当大小的缓存。当缓存中的分组数达到一定的数量后再以恒定速率按顺序把分组读出进行还原播放。缓存实际上就是一个先进先出的队列。图中标明的 T 叫做播放时延。



缓存使所有到达的分组都增加了迟延。早到达的分组在缓存中停留的时间较长，而晚到达的分组在缓存中停留的时间则较短。

以非恒定速率到达的分组，经过缓存后再以恒定速率读出，就能够在一定程度上消除了时延的抖动。



8.1 概述

需要解决的问题

在传送时延敏感的实时数据时，不仅传输时延不能太大，而且时延抖动也必须受到限制。

对于传送实时数据，很少量分组的丢失对播放效果的影响并不大（因为这是由人来进行主观评价的），因而是可以容忍的。丢失容忍也是实时数据的另一个重要特点。

由于分组的到达可能不按序，但将分组还原和播放时又应当是按序的。因此在发送多媒体分组时还应当给每一个分组加上序号。这表明还应当有相应的协议支持才行。

要使接收端能够将节目中本来就存在的正常的短时间停顿（如音乐中停顿几拍）和因某些分组的较大延迟造成的“停顿”区分开来。这就需要增加一个时间戳，以便告诉接收端应当在什么时间播放哪个分组。



8.1 概述

必须改造现有的因特网

大量使用光缆和高速路由器，网络的时延和时延抖动就可以足够小，在因特网上传送实时数据就不会有问题。把因特网改造为能够对端到端的带宽实现**预留**，把使用无连接协议的因特网转变为面向连接的网络。

部分改动因特网的协议栈所付出的代价较小，而这也能够使多媒体信息在因特网上的传输质量得到改进。

目前因特网提供的音频/视频服务可分为三种类型：

流式存储音频/视频：边下载边播放。

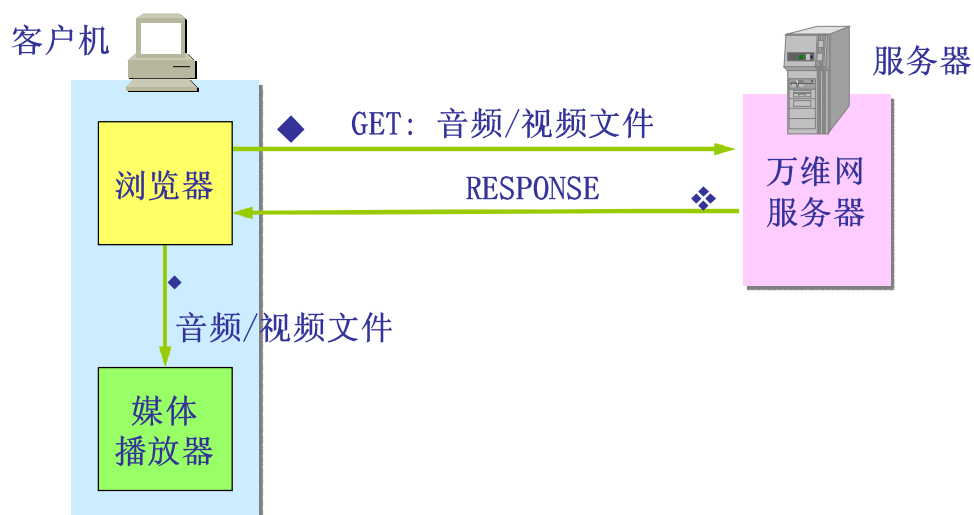
流式实况音频/视频：边录制边发送。

交互式音频/视频：实时交互式通信。



8.2 流式存储音频/视频

传统的下载文件方法





8.2 流式存储音频/视频

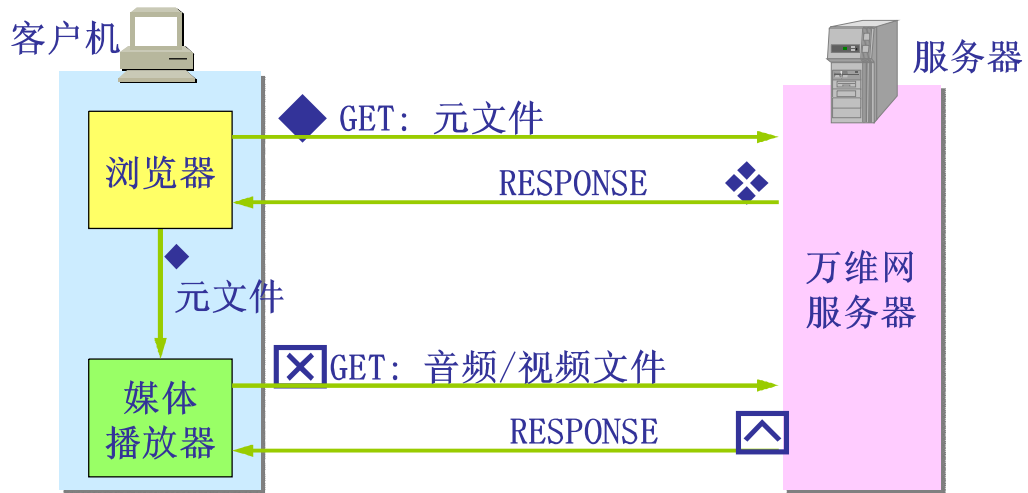
传统的浏览器从服务器下载音频/视频文件

- ① 用户从客户机(client machine)的浏览器上用HTTP 协议向服务器请求下载某个音频/视频文件。
- ② 服务器如有此文件就发送给浏览器。在响应报文中就装有用户所要的音频/视频文件。整个下载过程可能会花费很长的时间。
- ③ 当浏览器完全收下这个文件后，就可以传送给自己机器上的媒体播放器进行解压缩，然后播放。



8.2.1 具有元文件的万维网服务器

元文件就是一种非常小的文件，它描述或指明其他文件的一些重要信息。





8.2.1 具有元文件的万维网服务器

使用元文件下载音频/视频文件

① 浏览器用户使用HTTP的GET报文接入到万维网服务器。这个超链指向一个元文件。这个元文件有实际的音频/视频文件的统一资源定位符 URL。

② 万维网服务器把该元文件装入HTTP响应报文的主体，发回给浏览器。

③ 客户机浏览器调用相关的媒体播放器，把提取出的元文件传送给媒体播放器。

④ 媒体播放器使用元文件中的URL，向万维网服务器发送HTTP请求报文，要求下载音频/视频文件。

⑤ 万维网服务器发送HTTP响应报文，把该音频/视频文件发送给媒体播放器。播放器边下载边解压缩边播放。



8.2.2 媒体服务器

媒体服务器也称为流式服务器，它支持流式音频和视频的传送。

媒体播放器与媒体服务器是客户与服务器的关系。

媒体播放器不是向万维网服务器而是向媒体服务器请求音频/视频文件。

媒体服务器和播放器之间采用另外的协议进行交互。





8.2.2 媒体服务器

使用 RTSP 的媒体服务器的工作过程

- ① 浏览器向万维网服务器请求音频/视频文件。
- ② 万维网服务器从浏览器发送携带有元文件的响应。
- ③ 浏览器把收到的元文件传送给媒体播放器。
- ④ RTSP客户与媒体服务器的 RTSP 服务器建立连接。
- ⑤ RTSP服务器发送响应 RESPONSE 报文。
- ⑥ RTSP客户发送PLAY报文，开始下载音频/视频文件。
- ⑦ RTSP服务器发送响应 RESPONSE 报文。
- ⑧ RTSP客户发送 TEARDOWN 报文断开连接。
- ⑨ RTSP服务器发送响应 RESPONSE 报文。

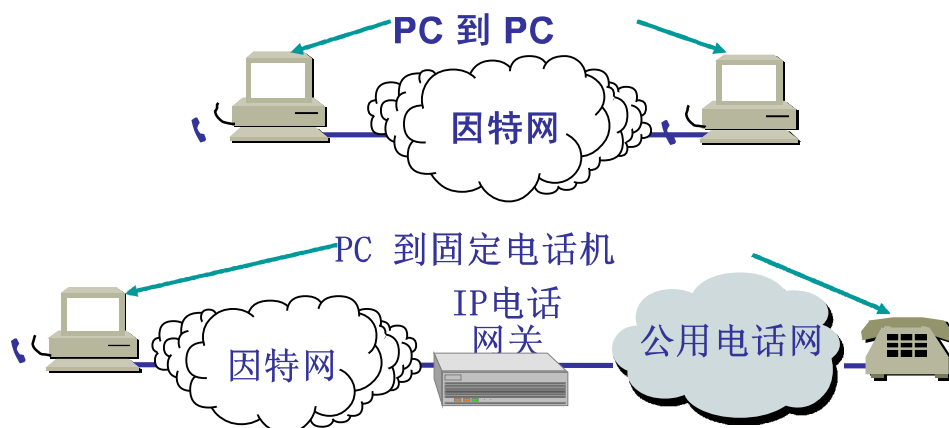


8.3 交互式音频/视频

8.3.1 IP电话概述

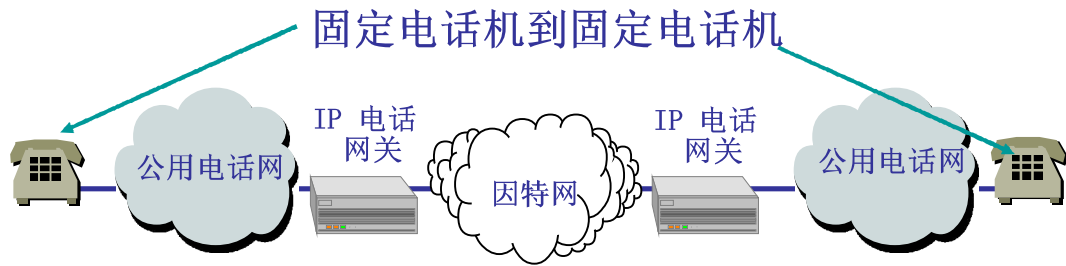
狭义的IP电话就是指在IP协议支持的网络上打电话。广义的IP电话则不仅仅是电话通信，而且还可以是在IP网络上进行交互式多媒体实时通信等。

IP电话网关的几种连接方法





8.3.1 IP电话概述



IP 电话的通话质量

IP电话的通话质量主要由两个因素决定。一是通话双方端到端的时延和时延抖动，另一个是话音分组的丢失率。但这两个因素是不确定的，是取决于当时网络上的通信量。

经验证明，在电话交谈中，端到端的时延不应超过250 ms，否则交谈者就能感到不自然。



8.3.1 IP电话概述

IP电话的端到端时延

- (1) 话音信号进行模数转换要经受时延。
- (2) 话音比特流装配成话音分组的时延。
- (3) 话音分组的发送需要时间，此时间等于话音分组长度与通信线路的数据率之比。
- (4) 话音分组在因特网中的存储转发时延。
- (5) 话音分组在接收端缓存中暂存所引起的时延。
- (6) 话音分组还原成模拟话音信号的时延。
- (7) 话音信号在通信线路上的传播时延。
- (8) 终端设备的硬件和操作系统产生的接入时延。

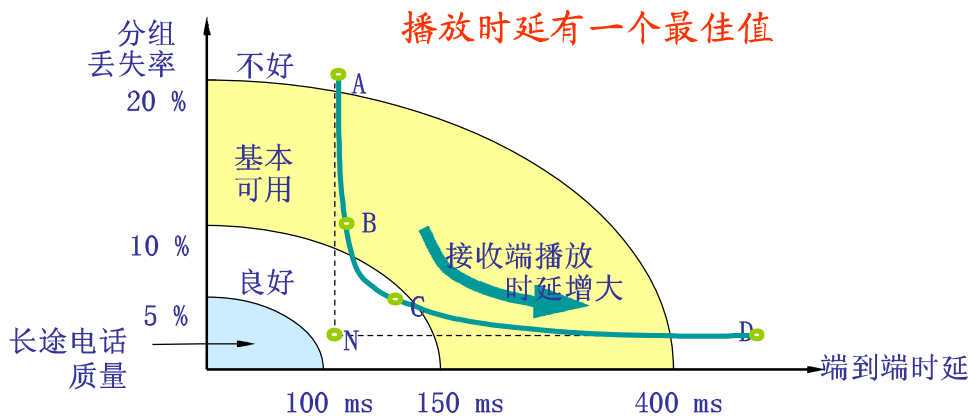


8.3.1 IP电话概述

低速率语音编码的标准

(1) G.729: 速率为8kb/s的共轭结构代数码激励线性预测声码器CS-ACELP。

(2) G.723.1: 速率为5.3/6.3kb/s的为多媒体通信用的低速率声码器。





8.3.1 IP电话概述

线速路由器

提高路由器的转发分组的速率对提高IP电话的质量也是很重要的。

据统计，一个跨大西洋的IP电话一般要经过 20~30个路由器。

若能改用吉比特路由器（线速路由器），则每秒可转发5百万至6千万个分组（即交换速率达60Gb/s 左右）。这样还可进一步减少由网络造成的时延。



8.3.1 IP电话概述

关于Skype

Skype 采用了P2P和全球索引技术提供快速路由选择机制，管理成本大大降低。由于用户路由信息分布式存储于因特网的结点中，因此呼叫连接完成得很快。

Skype 采用了端对端加密方式，保证信息的安全性。

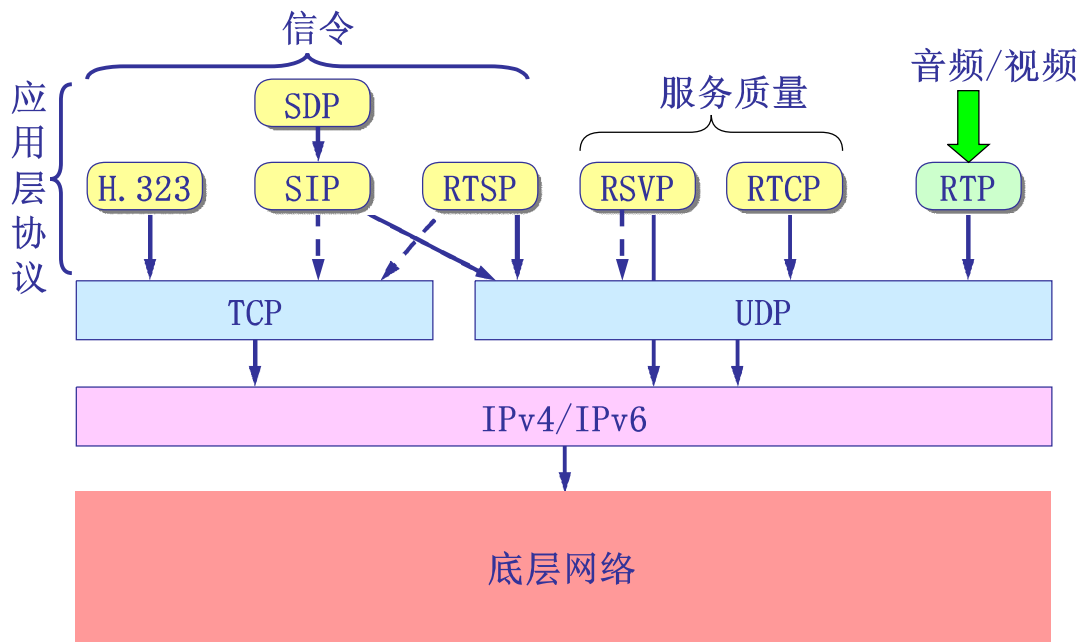
Skype 使用P2P的技术，用户数据主要存储在P2P网络中，因此必须保证存储在公共网络中的数据是可靠的和没有被篡改的。

Skype 对公共目录中存储的和用户相关的数据都采用了数字签名，保证了数据无法被篡改。

Skype的问世给全球信息技术和通信产业带来深远的影响，也给每一位网络使用者带来生活方式的改变。



8.3.2 IP电话所需要的几种应用协议





8.3.3 实时传输协议RTP

RTP为实时应用提供端到端的运输，但不提供任何服务质量的保证。

多媒体数据块经压缩编码处理后，先送给RTP封装成为RTP分组，再装入运输层的UDP用户数据报，然后再交给IP层。

RTP是一个协议框架，只包含了实时应用的一些共同的功能。

RTP自己并不对多媒体数据块做任何处理，而只是向应用层提供一些附加的信息，让应用层知道应当如何进行处



8.3.3 实时传输协议RTP

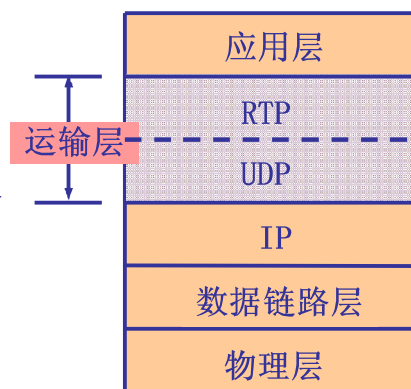
RTP的层次

从应用开发者的角度看，RTP应当是应用层的一部分。在应用的发送端，开发者必须编写用RTP封装分组的程序代码，然后把RTP分组交给UDP插口接口。

在接收端，RTP分组通过UDP插口接口进入应用层后，还要利用开发者编写的程序代码从RTP分组中把应用数据块提取出来。

RTP也可看成是运输层的子层

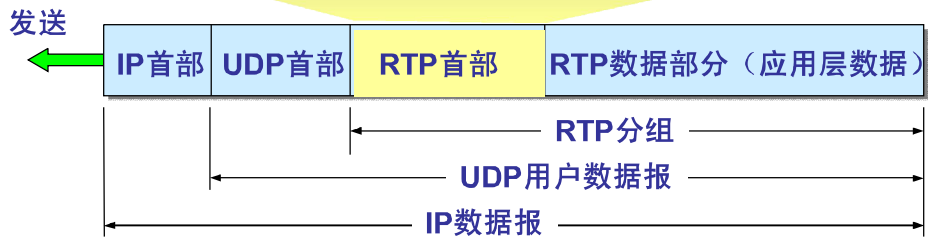
RTP封装了多媒体应用的数据块。由于RTP向多媒体应用程序提供了服务（如时间戳和序号），因此也可以将RTP看成是在UDP之上的一个运输层的子层。





8.3.3 实时传输协议RTP

RTP分组的首部格式





8.3.4 实时运输控制协议RTCP

RTCP 是与 RTP 配合使用的协议。

RTCP协议的主要功能是：服务质量的监视与反馈、媒体间的同步，以及多播组中成员的标识。

RTCP分组也使用UDP传送，但RTCP并不对声音或视像分组进行封装。

可将多个RTCP分组封装在一个UDP用户数据报中。

RTCP分组周期性地在网上传送，它带有发送端和接收端对服务质量的统计信息报告。



8.3.4 实时传输控制协议RTCP

RTCP 使用的五种分组类型

结束分组BYE表示关闭一个数据流。

特定应用分组APP使应用程序能够定义新的分组类型。

接收端报告分组RR用来使接收端周期性地向所有的点用多播方式进行报告。

发送端报告分组SR用来使发送端周期性地向所有接收端用多播方式进行报告。

源点描述分组SDES给出会话中参加者的描述。



8.3.5 H.323

H.323是ITU-T于1996年制订的一个名称很长的建议书，1998年的第二个版本改用的名称是“基于分组的多媒体通信系统”。

H.323包括系统和构件的描述，呼叫模型的描述，呼叫信令过程，控制报文，复用，话音编解码器，视像编解码器，以及数据协议等，但不保证服务质量 QoS。

H.323 终端使用 H.323 协议进行多媒体通信





H.323 标准指明的四种构件

(1) H.323 终端

(2) 网关：网关连接到两种不同的网络，使H.323网络可以和非H.323网络进行通信。

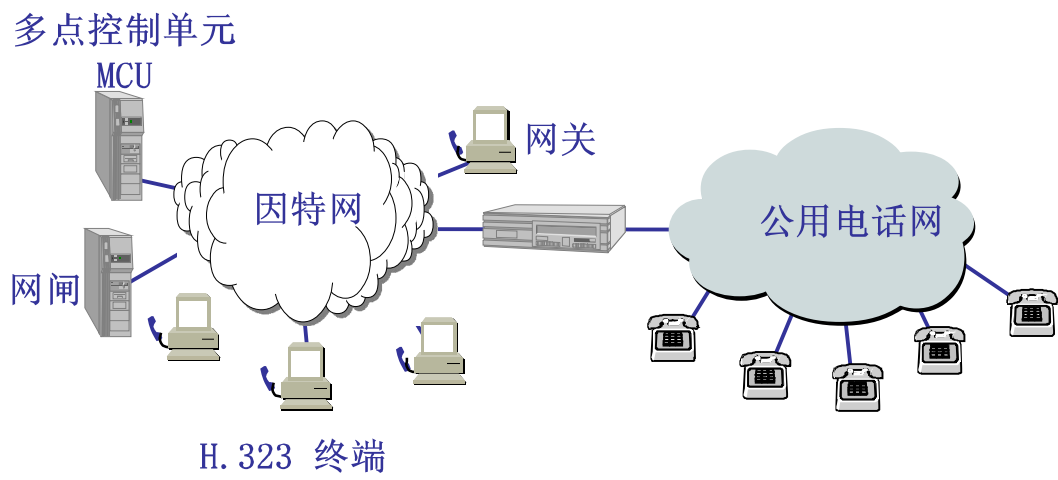
(3) 网闸：所有的呼叫都要通过网闸，因为网闸提供地址转换、授权、带宽管理和计费功能。

(4) 多点控制单元MCU：支持三个或更多的H.323终端的音频或视频会议。



8.3.5 H.323

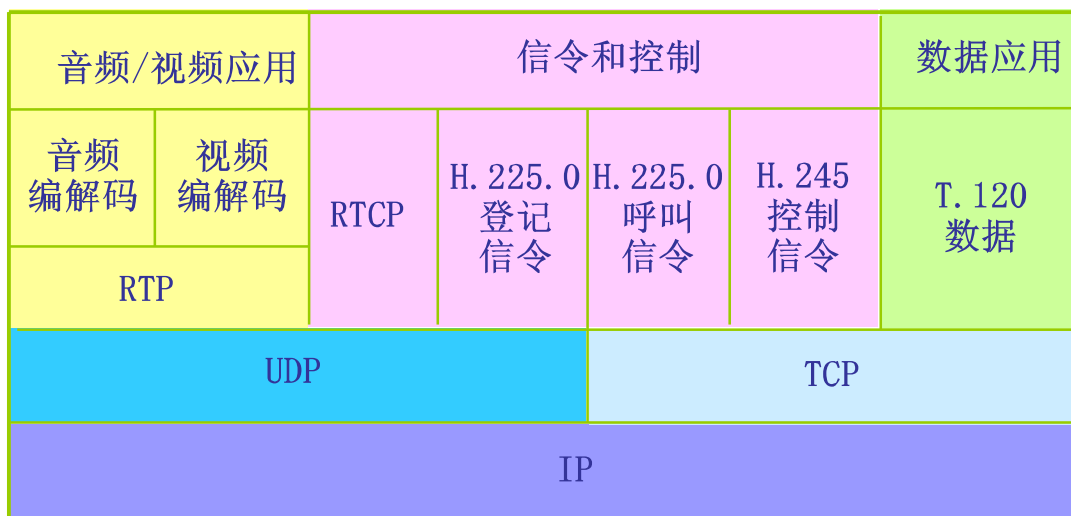
H.323网关用来和非H.323网络进行连接





8.3.5 H.323

H.323的协议体系结构





8.3.6 会话发起协议SIP

SIP是一套较为简单且实用的标准，目前已成为因特网的建议标准。

SIP协议以因特网为基础，把IP电话视为因特网上的新应用。

SIP协议只涉及到IP电话的信令和有关服务质量问题，而没有提供像H.323那样多的功能。

SIP没有指定使用 RTP 协议，但实际上大家还是选用 RTP 和 RTCP 作为配合使用的协议。



8.3.6 会话发起协议SIP

SIP系统的构件

SIP系统的两种构件是用户代理和网络服务器。

用户代理包括用户代理客户和用户代理服务器，前者用来发起呼叫，而后者用来接受呼叫。

网络服务器分为代理服务器和重定向服务器。

代理服务器接受来自主叫用户的呼叫请求，并将其转发给下一跳代理服务器，最后将呼叫请求转发给被叫用户。

重定向服务器不接受呼叫，它通过响应告诉客户下一跳代理服务器的地址，由客户按此地址向下一跳代理服务器重新发送呼叫请求。



8.3.6 会话发起协议SIP

SIP的地址十分灵活

可以是电话号码，也可以是电子邮件地址、IP地址或其他类型的地址。但一定要使用SIP的地址格式，例如：

电话号码： sip: zhangsan@8625-87654321

IPv4地址： sip: zhangsan@201.12.34.56

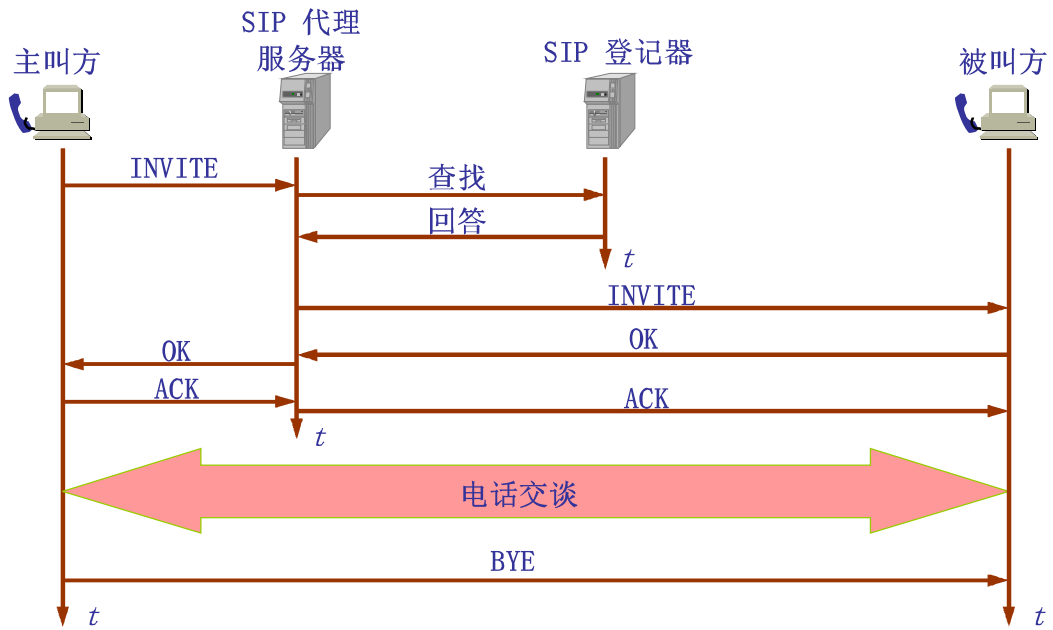
电子邮件地址： sip: zhangsan@public1.ptt.js.cn





8.3.6 会话发起协议SIP

SIP 登记器的用途: 跟踪被叫方





8.3.6 会话发起协议SIP

会话描述协议SDP

SDP在电话会议的情况下特别重要，因为电话会议的参加者是动态地加入和退出。

SDP详细地指明了媒体编码、协议的端口号以及多播地址。

SIP使用了HTTP的许多首部、编码规则、差错码以及一些鉴别机制，它比H. 323具有更好的可扩展性。

由于SIP问世较晚，因此它现在比H. 323占有的市场份额要小。



8.4 尽最大努力交付的服务

8.4.1 使因特网提供服务质量

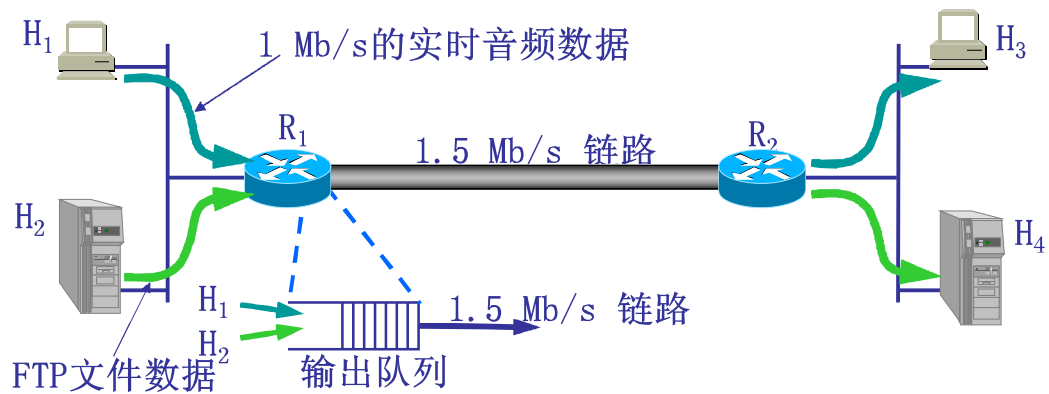
服务质量 QoS 是服务性能的总效果，此效果决定了一个用户对服务的满意程度。因此在最简单的意义上，有服务质量的服务就是能够满足用户的应用需求的服务。

服务质量可用若干基本的性能指标来描述，包括可用性、差错率、响应时间、吞吐量、分组丢失率、连接建立时间、故障检测和改正时间等。服务提供者可向其用户保证某一种等级的服务质量。



8.4.1 使因特网提供服务质量

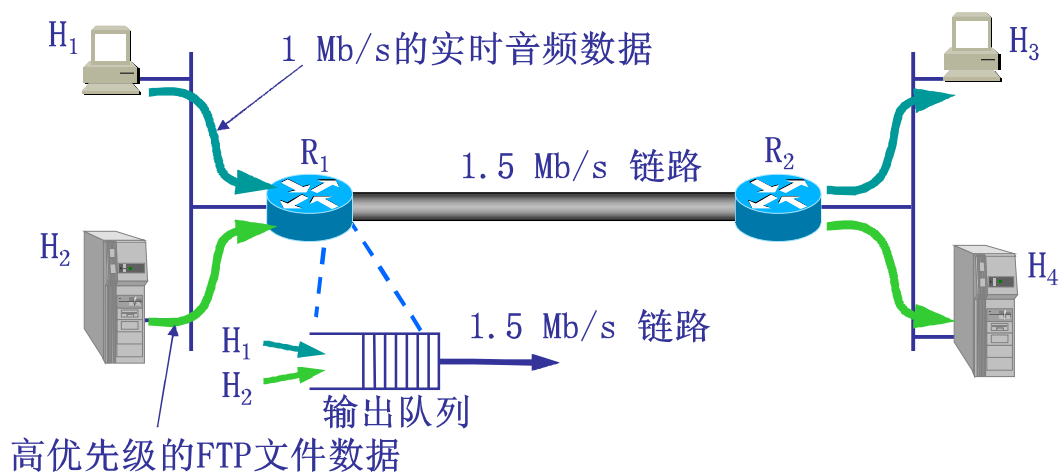
主机H1和H2分别向主机H3和H4发送数据



需要给不同性质的分组打上不同的标记。当H1和H2的分组进入R1时，R1应能识别实时数据分组，并使这些分组以高优先级进入输出队列，而仅在队列有多余空间时才准许低优先级的FTP数据分组进入。



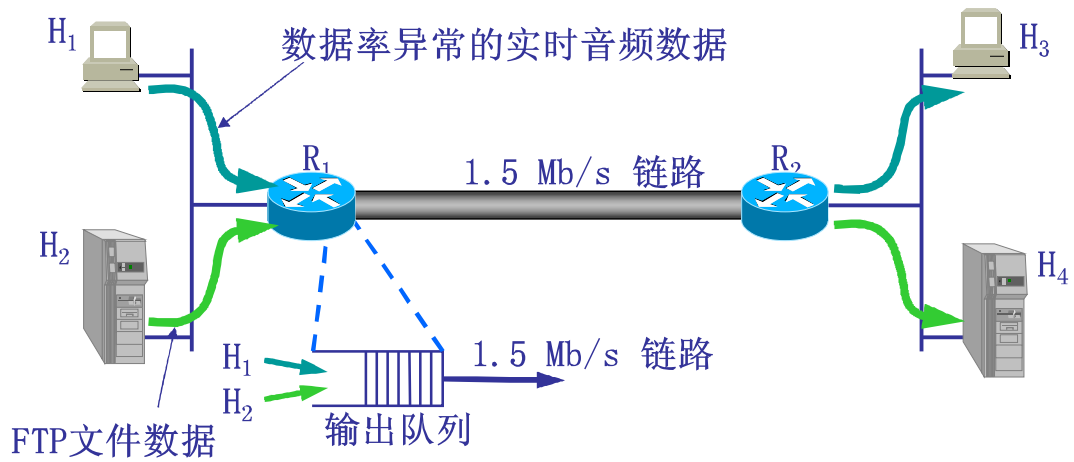
8.4.1 使因特网提供服务质量



应当使路由器增加分类(classification)机制，即路由器根据某些准则（例如，根据发送数据的地址）对输入分组进行分类，然后对不同类别的通信量给予不同的优先级。



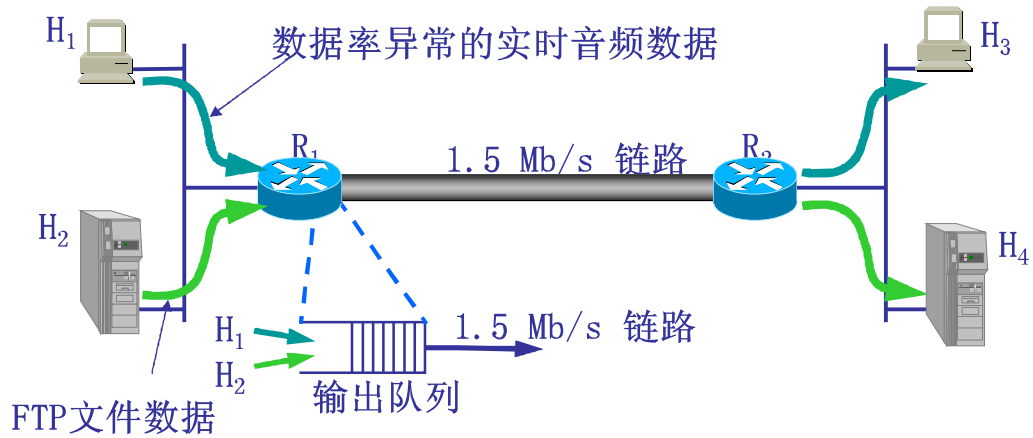
8.4.1 使因特网提供服务质量



路由器应能将数据流进行通信量的管制 (policing), 使该数据流不影响其他正常数据流在网络中通过。例如, 可将 H_1 的数据率限定为 1Mb/s 。 R_1 不停地监视 H_1 的数据率。只要其数据率超过规定的 1Mb/s , R_1 就将其中的某些分组丢弃。



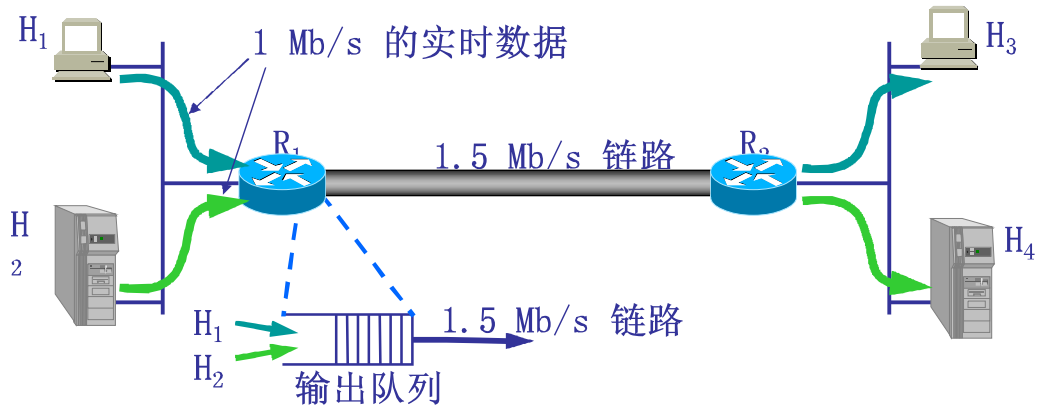
8.4.1 使因特网提供服务质量



应在路由器中再增加调度机制。利用调度功能给实时音频分配 1.0 Mb/s 的带宽，给文件传送分配 0.5 Mb/s 的带宽（相当于在带宽为 1.5 Mb/s 的链路中划分出两个逻辑链路），因而对这两种应用都有相应的服务质量保证。



8.4.1 使因特网提供服务质量



总数据率已超过了1.5 Mb/s 链路的带宽。比较合理的做法是让一个数据流通过 1.5 Mb/s 的链路，而阻止另一个数据流的通过。这就需要呼叫接纳 (call admission) 机制。数据流要预先声明所需的服务质量，然后或者被准许进入网络，或者被拒绝进入网络。



8.4.2 调度和管制机制

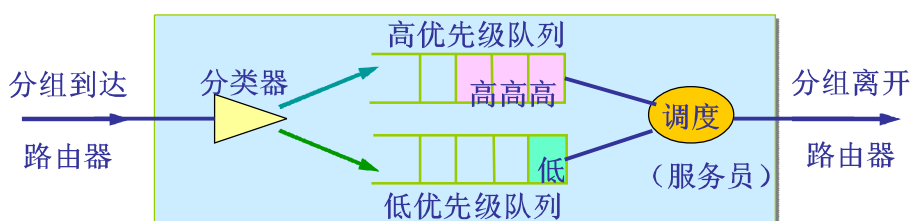
(1) 调度机制

“调度”就是指排队的规则。

如不采用专门的调度机制，则默认排队规则就是先进先出 FIFO (First In First Out)。当队列已满时，后到达的分组就被丢弃。

先进先出的最大缺点就是不能区分时间敏感分组和一般数据分组，并且也不公平。

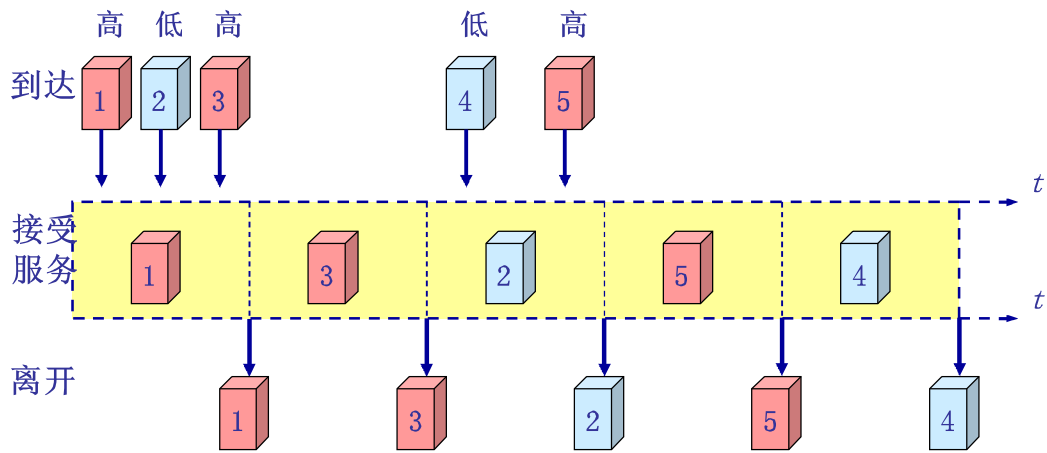
在先进先出的基础上增加按优先级排队，就能使优先级高的分组优先得到服务。





8.4.2 调度和管制机制

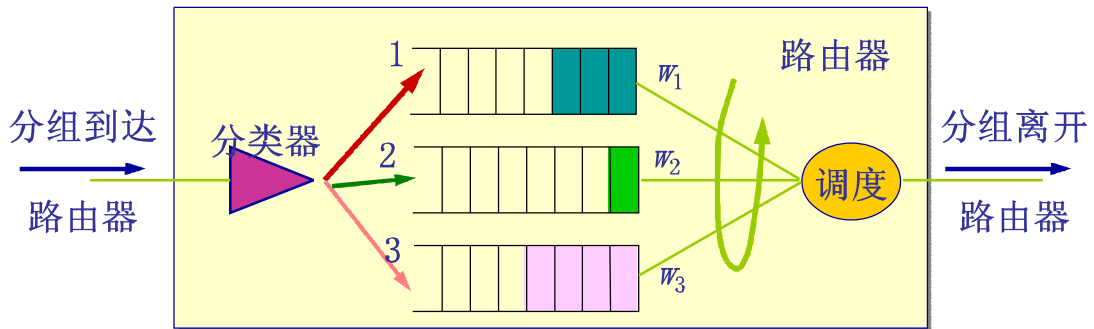
高优先级分组优先接受服务





8.4.2 调度和管制机制

加权公平排队 WFQ (Weighted Fair Queuing)



分组到达后就将分组进行分类，然后送交与其类别对应的队列。队列按顺序依次将队首的分组发送到链路。遇到队列空就跳过去。给队列 i 指派一个权重 w_i 。队列 i 得到的平均服务时间为 $w_i / (\sum w_j)$ ，这里 $\sum w_j$ 是对所有的非空队列的权重求和。队列 i 将得到的有保证的带宽

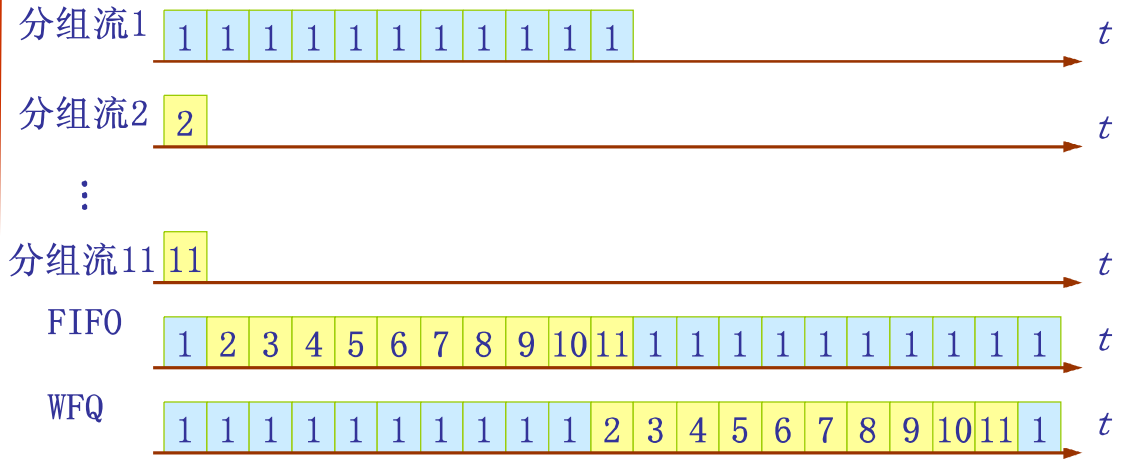
$$R_i = \frac{R \times w_i}{\sum w_j}$$



8.4.2 调度和管制机制

WFQ与FIFO的比较

(a) 分组流1的分组连续输入



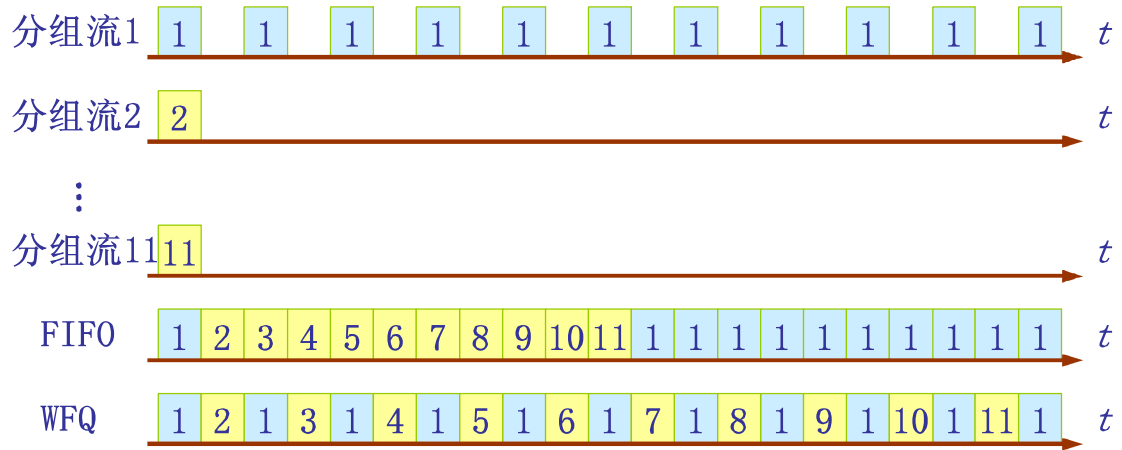


8.4.2 调度和管制机制



WFQ与FIFO的比较

(b) 分组流1的分组断续输入





8.4.2 调度和管制机制

(2) 管制机制

平均速率：网络需要控制一个数据流的平均速率。这里的平均速率是指在一定的时间间隔内通过的分组数。

峰值速率：峰值速率限制了数据流在非常短的时间间隔内的流量。

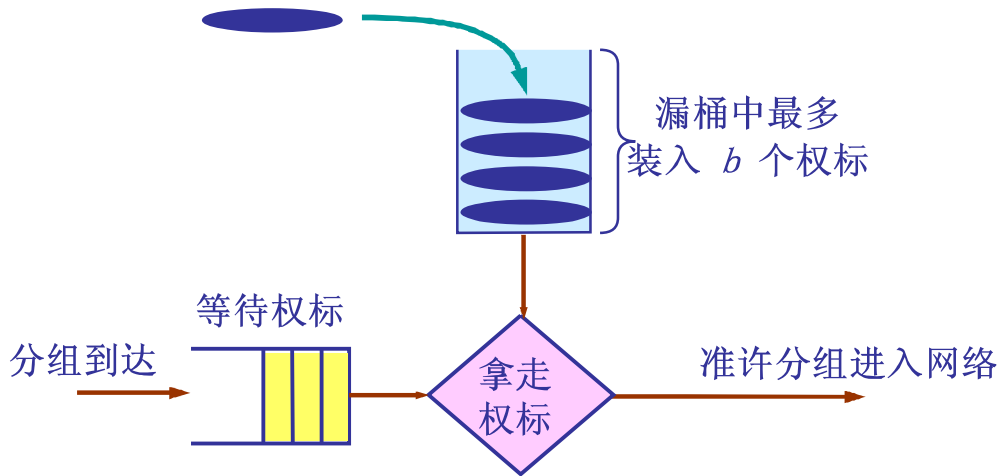
突发长度：网络也限制在非常短的时间间隔内连续注入到网络中的分组数。



8.4.2 调度和管制机制

漏桶管制器

标记注入漏桶的速率为每秒 r 个权标



在任何时间间隔 t 内准许进入网络的分组数 $= r t + b$



8.4.2 调度和管制机制

(3) 漏桶机制与加权公平排队相结合

现假定有 n 个分组流输入到一个路由器，复用后从一条链路输出。每一个分组流使用漏桶机制进行管制，漏桶参数为 b_i 和 r_i , $i = 1, 2, \dots, n$ 。

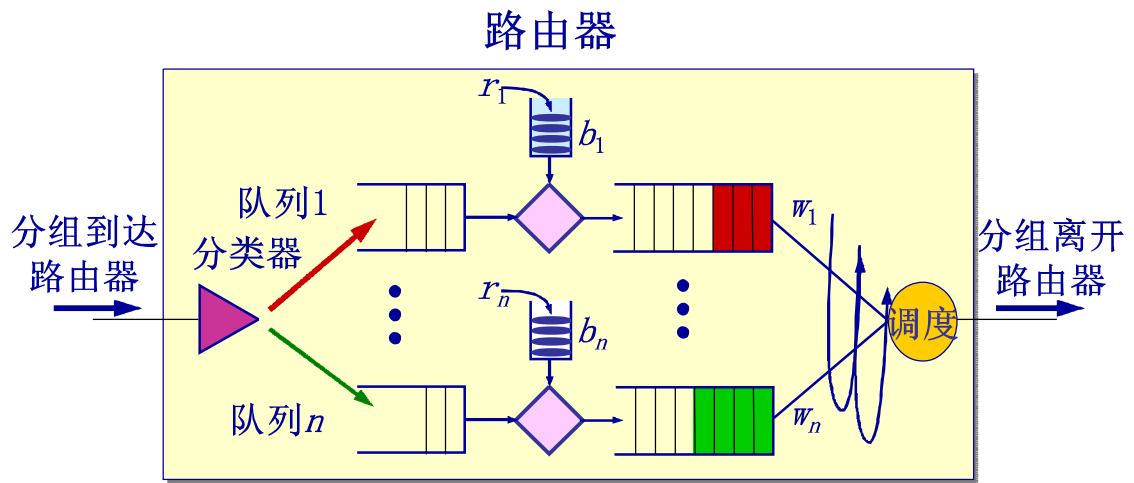
设漏桶 I 已装满了 b_i 个权标。因此 b_i 个分组可马上从路由器输出。但分组流 I 得到的带宽是由公式(10-1)给出。这 b_i 个分组中的最后一个分组所经受的时延最大，它等于传输这 b_i 个分组所需的时间 d_{\max} ，即 b_i 除以公式(10-1)给出的传输速率：

$$d_{\max} = \frac{b_i}{R \times w_i / \sum w_j}$$



8.4.2 调度和管制机制

用漏桶机制进行管制





8.4.3 综合服务与资源预留协议

IntServ (Integrated Services) 可对单个的应用会话提供服务质量的保证，其主要特点有二，即：

资源预留：路由器需要知道不断出现的会话已预留了多少资源（即链路带宽和缓存空间）。

呼叫建立：需要服务质量保证的会话必须首先在源站到目的站的路径上的每个路由器预留足够的资源，以保证其端到端的服务质量要求。

IntServ 定义了两类服务

有保证的服务 (guaranteed service)，可保证一个分组在通过路由器时的排队时延有一个严格的上限。

受控负载的服务 (controlled-load service)，可以使应用程序得到比通常的“尽最大努力”更加可靠的服务。



8.4.3 综合服务与资源预留协议

IntServ由四个组成部分

- (1) 资源预留协议 RSVP，它是 IntServ 的信令协议。
- (2) 接纳控制，用来决定是否同意对某一资源的请求。
- (3) 分类器，用来将进入路由器的分组进行分类，并根据分类的结果将不同类别的分组放入特定的队列。
- (4) 调度器：根据服务质量要求决定分组发送的顺序。

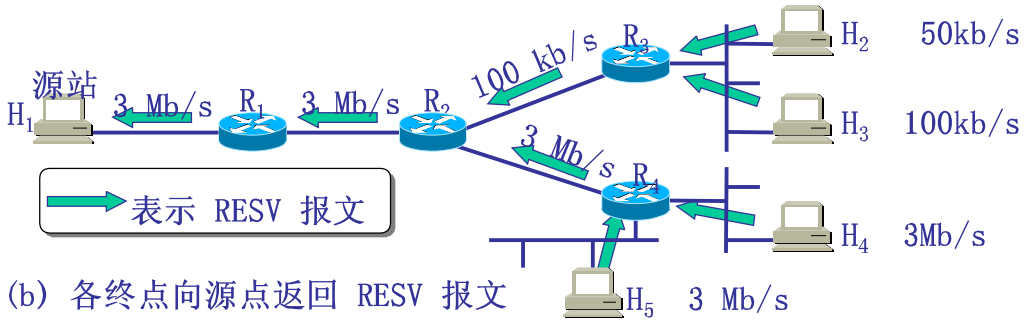
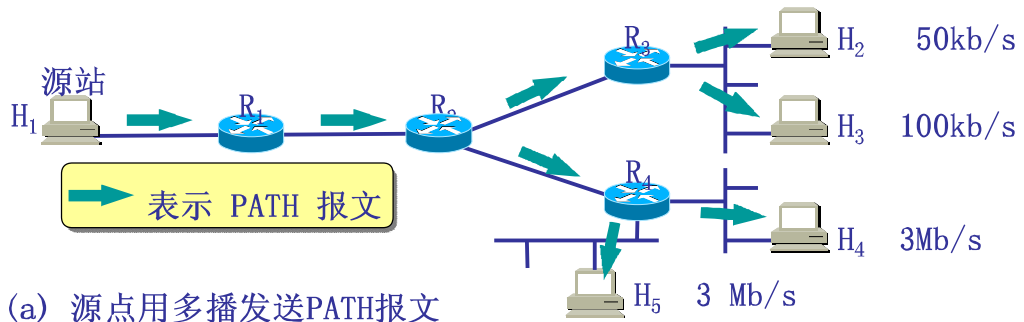
流(flow)

“流”是在多媒体通信中的一个常用的名词，一般定义为：具有同样的源IP地址、源端口号、目的IP地址、目的端口号、协议标识符以及服务质量需求的一连串分组。



8.4.3 综合服务与资源预留协议

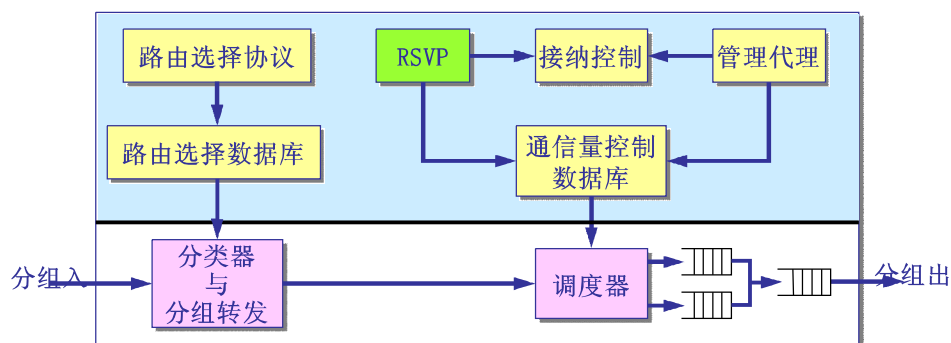
RSVP 协议的工作原理





8.4.3 综合服务与资源预留协议

IntServ体系结构在路由器中的实现



综合服务IntServ体系结构存在的主要问题

(1) 状态信息的数量与流的数目成正比。因此在大型网络中，按每个流进行资源预留会产生很大的开销。

(2) IntServ体系结构复杂。若要得到有保证的服务，所有的路由器都必须装有 RSVP、接纳控制、分类器和调度器。

(3) IntServ所定义的服务质量等级数量太少，不够灵活。



8.4.4 区分服务DiffServ

(1) 区分服务的基本概念

由于综合服务IntServ和资源预留协议RSVP都较复杂，很难在大规模的网络中实现，因此IETF提出了新的策略，即区分服务DiffServ。

区分服务有时也简称为DS。因此，具有区分服务功能的结点就称为DS结点。

(2) 区分服务DiffServ的要点

(1) DiffServ在路由器中增加区分服务的功能。

DiffServ将IPv4协议中原有的服务类型字段和IPv6的通信量类字段定义为区分服务字段DS。路由器根据DS的值来转发分组。利用DS可提供不同等级的服务质量。DS现只使用前6bit，即**区分服务码点** DSCP。

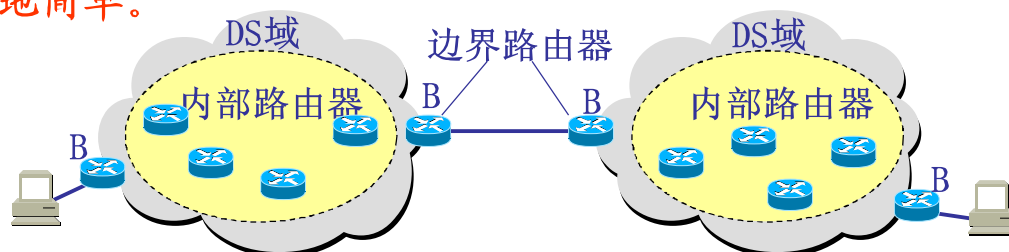


8.4.4 区分服务DiffServ

(3) 服务等级协定SLA

在使用DS字段之前，因特网的ISP要和用户商定一个服务等级协定SLA。在SLA中指明了被支持的服务类别（可包括吞吐量、分组丢失率、时延和时延抖动、网络的可用性）和每一类所容许的通信量。

网络被划分为许多个DS域，DiffServ将所有的复杂性放在DS域的边界结点中，而使DS域内部路由器工作得尽可能地简单。



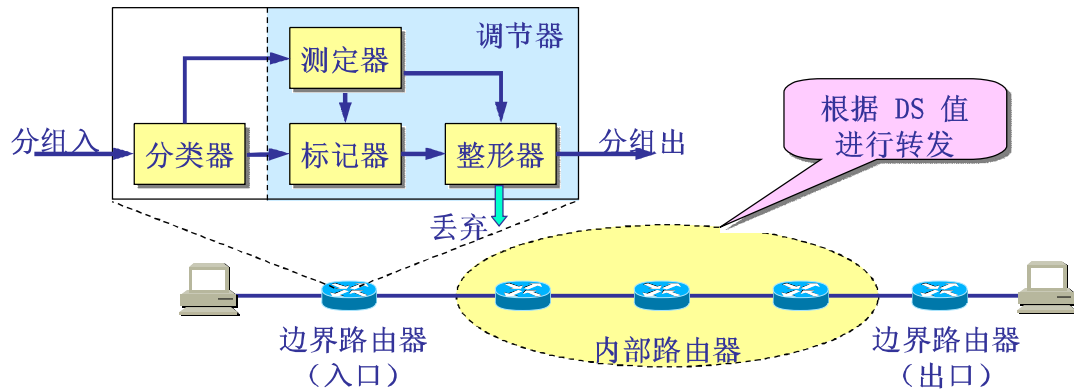


8.4.4 区分服务DiffServ

(4) 边界路由器中的功能

边界路由器中的功能较多，可分为：分类器和通信量调节器两大部分。其中调节器由标记器、整形器和测定器组成。

边界路由器中的各功能块的关系





8.4.4 区分服务DiffServ

(5) 聚合 (aggregation)

DiffServ 提供了一种聚合功能。

DiffServ不是为网络中的每一个流维持供转发时使用的状态信息，而是将若干个流根据其DS值聚合成少量的流。

路由器对相同DS值的流都按相同的优先级进行转发。这就大大简化了网络内部的路由器的转发机制。

区分服务DiffServ 不需要使用RSVP 信令。

(6) 每跳行为PHB

“行为”是指在转发分组时路由器是怎样处理分组的。

“每跳”是强调这里所说的行为只涉及到本路由器转发的这一跳，而下一路由器怎样处理则与本路由器处理无关。这和IntServ/RSVP考虑的“端到端”服务质量的很不一样。



8.4.4 区分服务DiffServ

(7) DiffServ定义的两类PHB

迅速转发PHB (EF PHB或EF) : 指明离开一个路由器的通信量的数据率必须等于或大于某一数值。因此EF PHB用来构造通过DS域的低丢失率、低时延、低时延抖动、确保带宽的端到端服务。像点对点连接或“虚拟租用线”，又称为Premium服务。

确保转发PHB (AF PHB或AF) : AF用DSCP的比特0-2将通信量划分为四个等级，并给每一种等级提供最低数量的带宽和缓存空间。

对于其中的每一个等级再用DSCP的比特3-5划分出三个“丢弃优先级”。当发生网络拥塞时，对于每个等级的AF，路由器首先把“丢弃优先级”较高的分组丢弃。



第8章 习题

8.01, 8.02, 8.03, 8.04, 8.05,
8.06, 8.07, 8.08, 8.09, 8.10,
8.12, 8.13, 8.14, 8.16, 8.17,
8.18, 8.19, 8.20, 8.22, 8.28,
8.29, 8.50, 8.31。