

2.9 服务质量-QoS

- ◆ 背景：当前网络是 best-effort model，核心是FIFO，网络传送数据并不作任何许诺
- ◆ 需求：实时应用需要来自网络的某种保证，即数据应按时/及时到达，即根据不同需求提供不同服务质量-Quality of Service
 - ♣ 实时多媒体流传输
 - ♣ 主动/互动网络命令传输

QoS的不同要求

◆ V4/FIFO Best effort 机制 (本章计划学时-6H)

◆ 不同多媒体在网上传输有不同要求

- ♣ E-Mail需要100%正确性，不强调速度
- ♣ Telnet需要实时可靠正确，传输量不大
- ♣ 音频允许丢失和一定延迟，不允许间隔和干扰
- ♣ 视频允许丢失，但要延迟小吞吐率高、抖动

因特网应用的性能要求

	电子邮件	Telnet	Iphone	组播视频通信
延迟	几分~几小时	几分~几秒	几秒	毫秒~秒
丢失率	0%	0%	小于 25%	允许/平均分布
连续性	无要求	可等待	连续	连续
同步	无要求	无要求	无	有
抖动	无要求	无要求	无串音	不允许

IETF关于QoS的两种框架

◆ IntServ(Integrated Service)

- ♣ 为每个流预留资源，可实现端到端的QoS
- ♣ 缺点是可扩展性差

◆ DiffServ(Differentiated Service)

- ♣ 根据IP包头将数据包分类，可扩展性好，适合大规模网络的应用
- ♣ DiffServ可细分为两类
 - ☞ 绝对服务区分
 - ☞ 相对服务区分，实现简单，可扩展性好，发展出比例QoS

解决办法

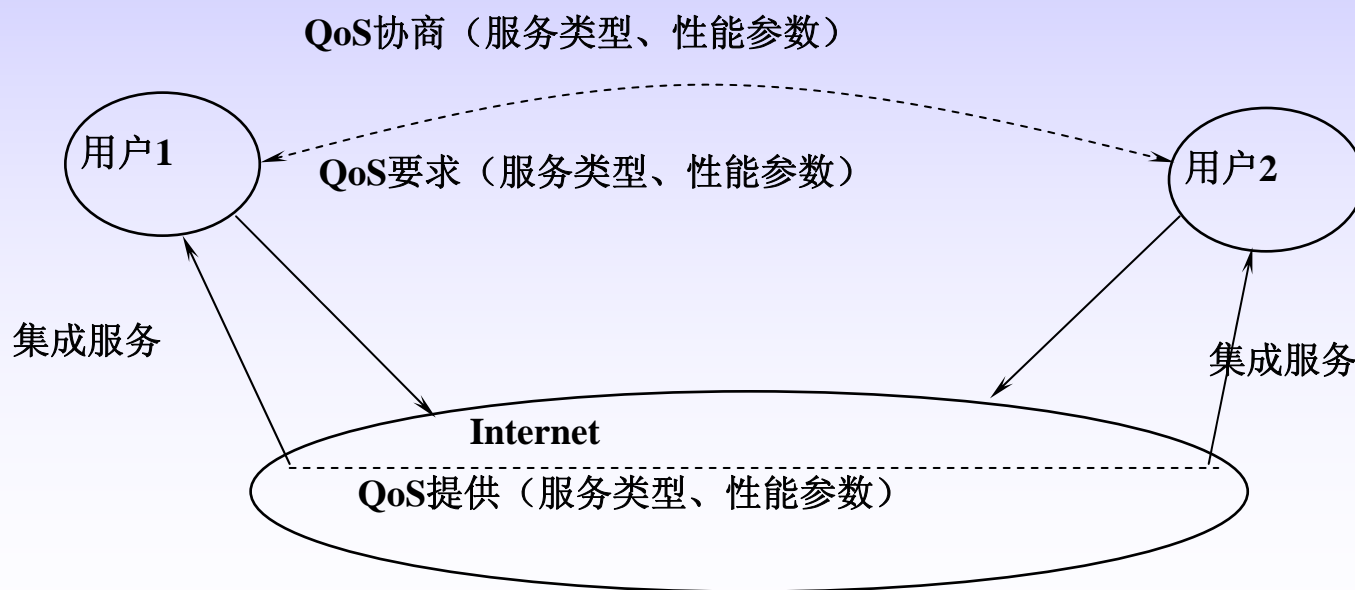
- ◆ 建立全新体系结构：如电视电话优先
 - ♣ 但这失去了实时/非实时应用统计复用网络资源，提高利用率的优点
 - ♣ 因特网上广大用户已习惯Internet通信环境
- ◆ 在Internet基础上开发实时/非实时应用
 - ♣ '90实验Mbone，当网上无其它应用共享带宽时，音/视有很好性能
 - ♣ 一旦有其它应用，Mbone质量迅速下降

怎样解决集成服务？

- ◆ 集成服务：D/V/V Best effort
 - ♣ 增加现有网络的带宽
 - ♣ 在R中增加对各种不同流量的优先级管理
 - ♣ 由应用系统用自适应方式抢占网络资源
- ◆ IETF认为上述办法都不能解决问题
- ◆ 事实是：资源总是有限的，需求则是无限的，仅靠R上优先级无法解决
- ◆ **机制/端到端角度：按用户 Qos和网络当时能力进行协商**

服务质量标准QoS

- ◆ QoS定义：发和收用户间、用户与网络间关于信息传输质量的约定
- ◆ 质量包括：用户要求+集成服务提供者的行为



QoS是系统性能表现

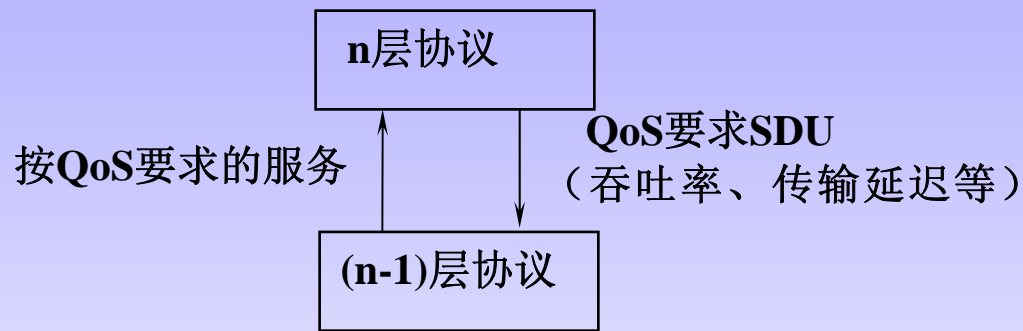
- ◆ 用户与用户/用户与网络/网元与网元
- ◆ 户A30/s帧—户B20/s帧，尽管网能提供30/s, 但户B接收不力而丢帧
- ◆ 降低户A、网的速度到20/s，双赢
- ◆ 网络无法满足用户要求时，用户和系统间进行QoS协商→准入控制
- ◆ Qos引入是对因特网机制的重大改变

OSI PR中的QoS定义

◆ 高层向低层发QoS参数，低层按高层要求进行QoS操作

参数	含义
吞吐量	单位时间内在一个连接上传递的最大字节数
传输延迟	从数据传输请求开始到数据传输完成确认为止的时间间隔
出错率	数据单元错传、丢失或重传的概率
建立连接延迟	从请求建立连接开始到建立连接确认为止的时间间隔
连接失败率	建立连接失败的概率
传输失败率	传输失败的概率
重置率	在给定时间内服务提供者
释放延迟	从释放请求开始到释放确认为止的时间延迟
释放失败概率	释放连接时失败的概率

与协议无关的QoS参数



参数	含义
访问权限	防止非法用户访问
优先级	包括传输优先级和使用优先级
成本	信息传输时所消耗的资源或资金

ITU的QoS定义

- ◆ 主要针对数据传输
- ◆ OSI未给出实行QoS的方法与框架
- ◆ 呼叫控制：包括呼叫次数、失败率
- ◆ 连接级：连接延迟/失败，释放延迟/失败
- ◆ 数据单元传输控制：报文峰值到达率、峰值持续时间、报文平均到达率/丢失率/插入率及比特出错率
- ◆ 仍未实现QoS控制和机制

ATM的QoS定义

- ◆ QoS 引入到ATM交换机，并对应用和服务分类
- ◆ 服务定义为5类：
 - ♣ CBR/rt-VBR/nrt-VBR/ABR/UBR
- ◆ 定义呼叫准入控制CAC
 - ♣ 检查用户请求并根据资源决定接受或拒绝连接请求
- ◆ ATM把UNI服务类型匹配成QoS参数->不同信元->不同优先级对列

IETF的QoS定义

◆ IETF把QoS控制分为两部分

- ♣ 集成服务模型：RF1633
- ♣ QoS实现框架

◆ 相继RFC

- ♣ 因特网控制负载型服务：RFC2211
- ♣ 保证型服务：RFC2212
- ♣ 集成服务（通过QoS控制参数 ） RFC2215
- ♣ QoS服务规范：2216
- ♣ QoS控制的关键/资源预约RSVP：RF2205

QoS的定义

◆ **QoS** 定义：RFC2216:

♣ QoS是用带宽、分组延迟、和分组丢失率等参数描述的关于分组传输的质量

◆ **网络元素**:

♣ 可在Internet中处理数据报文的构件，它在数据通过时具有进行QoS控制的能力，包括R、子网、端主机系统的OS等

◆ **流**: 有相同QoS要求和服从同一QoS控制方法的通过某个网络元素的报文集合，给定网络中，流可能来自不同应用

QoS的定义

- ◆ **服务**: 与QoS具有相同的意义, 含规范和功能; 两大部分。
- ◆ **行为**: 与QoS相关的端到端的性能, 是应用直接可见的最终结果
- ◆ **流量规范Tspec**: 要求服务提供的流量描述, 实际上是一份数据流和网络元素提供的服务之间的合同
- ◆ 还定义了**服务规范**、其它**控制语句**等

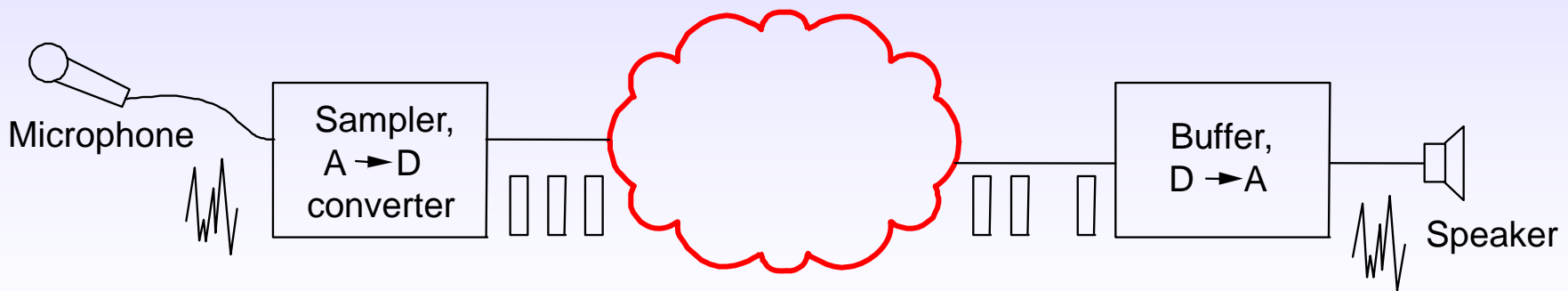
2.9.1 应用需求

◆应用基本分类:

- ♣非实时:传统应用:Telnet, FTP, email, Web..
不需保证数据的实时性, 延迟小当然好, 延迟大也可以用, 弹性数据
- ♣实时:对数据及时性敏感的应用:实时音频, 视频, 主动/互动命令数据

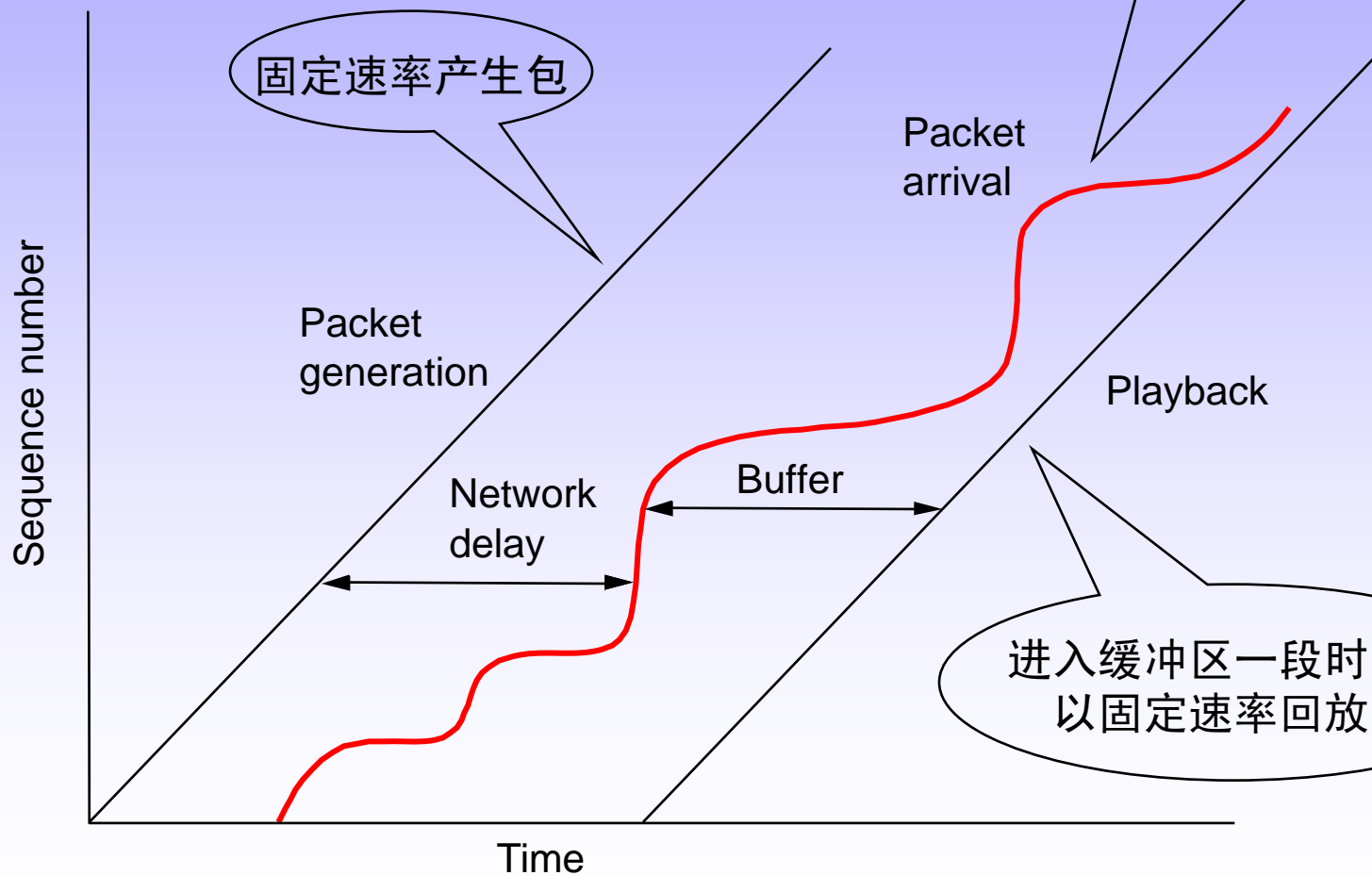
A: 实时音频应用的例子

- ◆ Macphone采样生成数据, A/D转换将数字化样本放在包中通过网络传输
- ◆ 接收端以125us接收, 并需以相当速率(125us)回放(playback), 故认为每个样本都有一个特定回放时刻
- ◆ 某个样本晚到时间过长会使它立刻变得无用

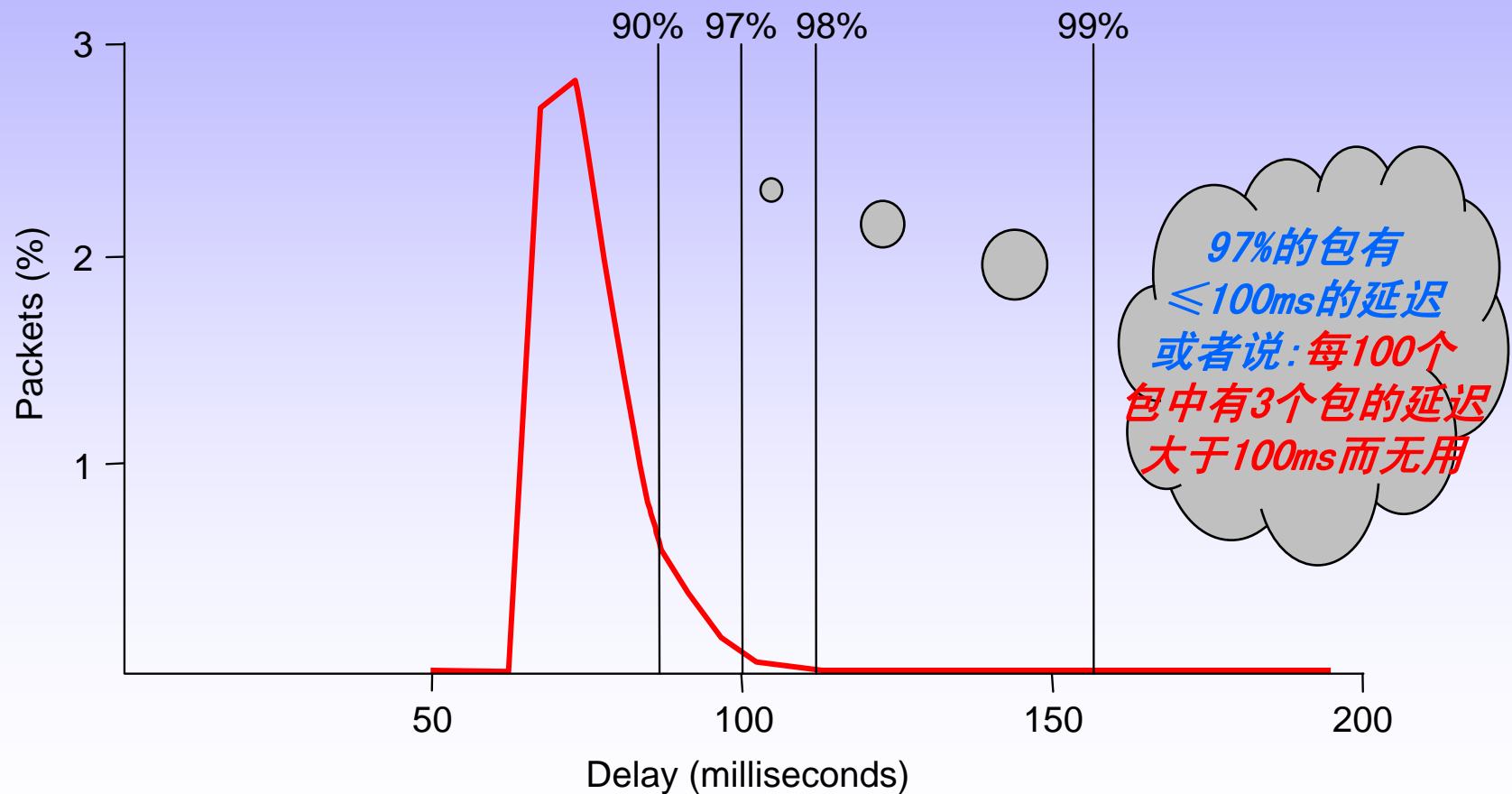


回放缓冲区

包到达的时间取决于
它在网络中遇到的情况
对音频延迟超过300ms
谈话就很难继续下去了



一个因特网连接延迟分布的例子



B: 实时应用的分类

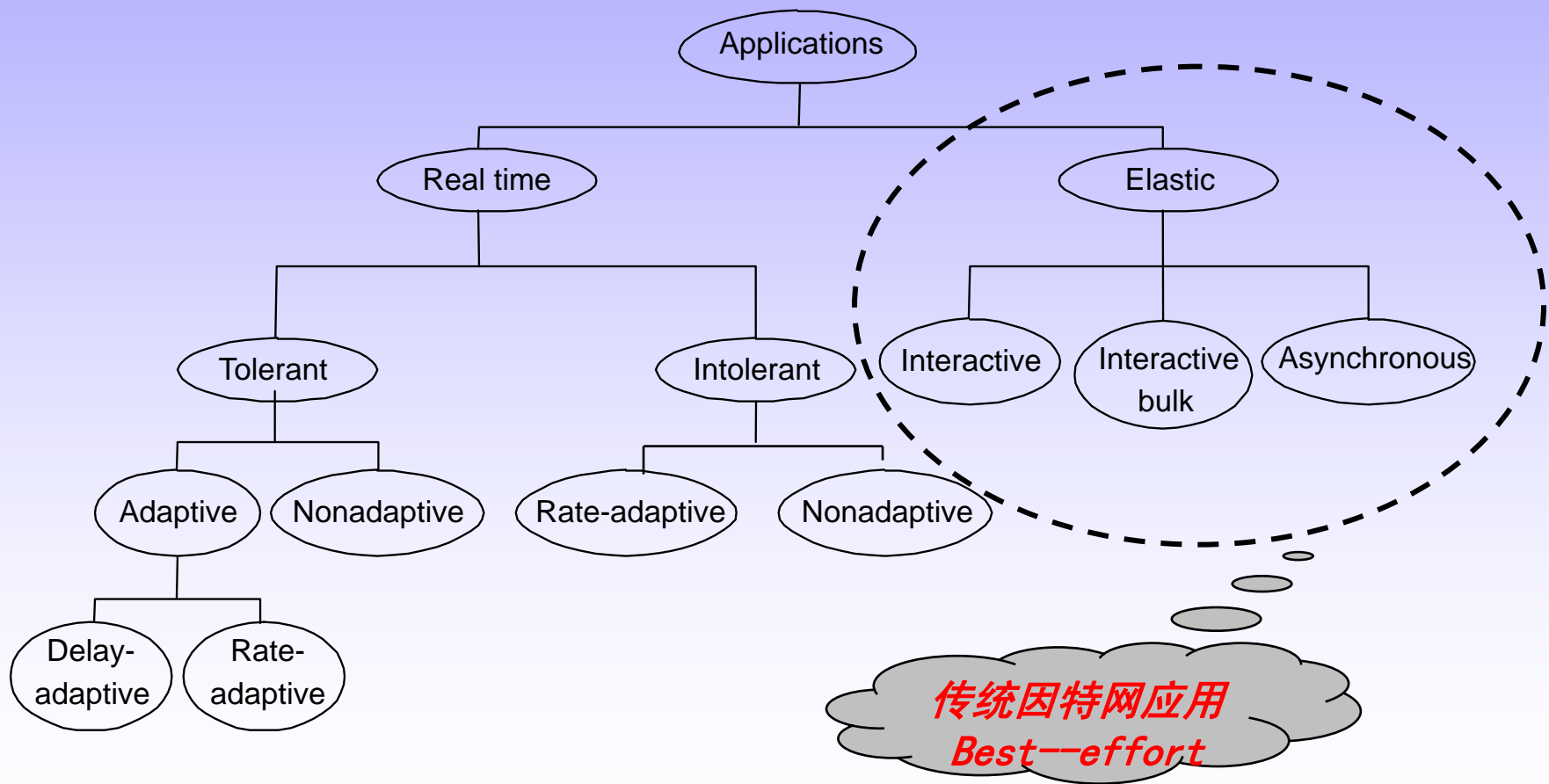
◆ 划分原则1：丢失数据的容忍度

- ♣ 容错型：音频/视频丢失1帧可通过插值等方法得到
- ♣ 非容错型：控制机器人手臂的命令丢失是不可接受的，还有银行数据等

◆ 划分原则2：适应性

- ♣ 延迟可适应的应用：一个音频应用可能适应包通过网络经历的延迟。若包总是在发送后300ms到达，则可响应设置回放点；若包总是在发送后100ms到达，则可将回放点提前到100ms，用户会感到改进
- ♣ 速率可适应的应用：许多视频编码算法能权衡比特率与质量，能根据网络带宽设置编码参数

应用的分类



C:支持QoS 的两类方法

- ◆ **细粒度方法：** 给单独的应用程序或流提供QoS。
 - ♣ **综合服务**—IETF开发的体系结构，常常与RSVP相关
 - ♣ **ATM**提供丰富的QoS功能，资源与单个虚电路有关
- ◆ **粗粒度方法：** 给多类数据或成块通信量提供 QoS。区别服务—IETF正在标准化

2.9.2 综合服务(RSVP)

- ◆ 95-97年间IETF综合服务工作组开发了大量的服务类规范来满足一些应用类的需要

(1) 服务类:

- ♣ 有保证服务:为非容忍应用设计,要求包永远不迟到,分配缓冲给早到的包,能设定回放点
- ♣ 受控负载服务:网络负荷不重时运行很好,如音频应用当包丢失率小于10%能产生合理音频质量,用WFQ把其它通信量同受控通信量分开
- ♣ 上述只是服务类型的一个子集,能否满足各类应用还待观察

(2) 机制描述

- ◆ 流说明flowspec: 尽力服务只能告诉网络我们想让包去哪里并把它留在哪里; 实时服务要求网络提供服务类型的信息
- ◆ 准入控制: 向网络请求特殊服务时, 网络要判断是否能提供这种服务. 10用户都请求2Mbps而他们都共享10Mbps, 故**不得不拒绝**某些用户
- ◆ 资源预定: 用户与网元间交换信息, 如请求服务, 流说明及准入控制的决定
- ◆ 包调度, R和S要满足流的需求管理包队列并调度

(3) 流说明

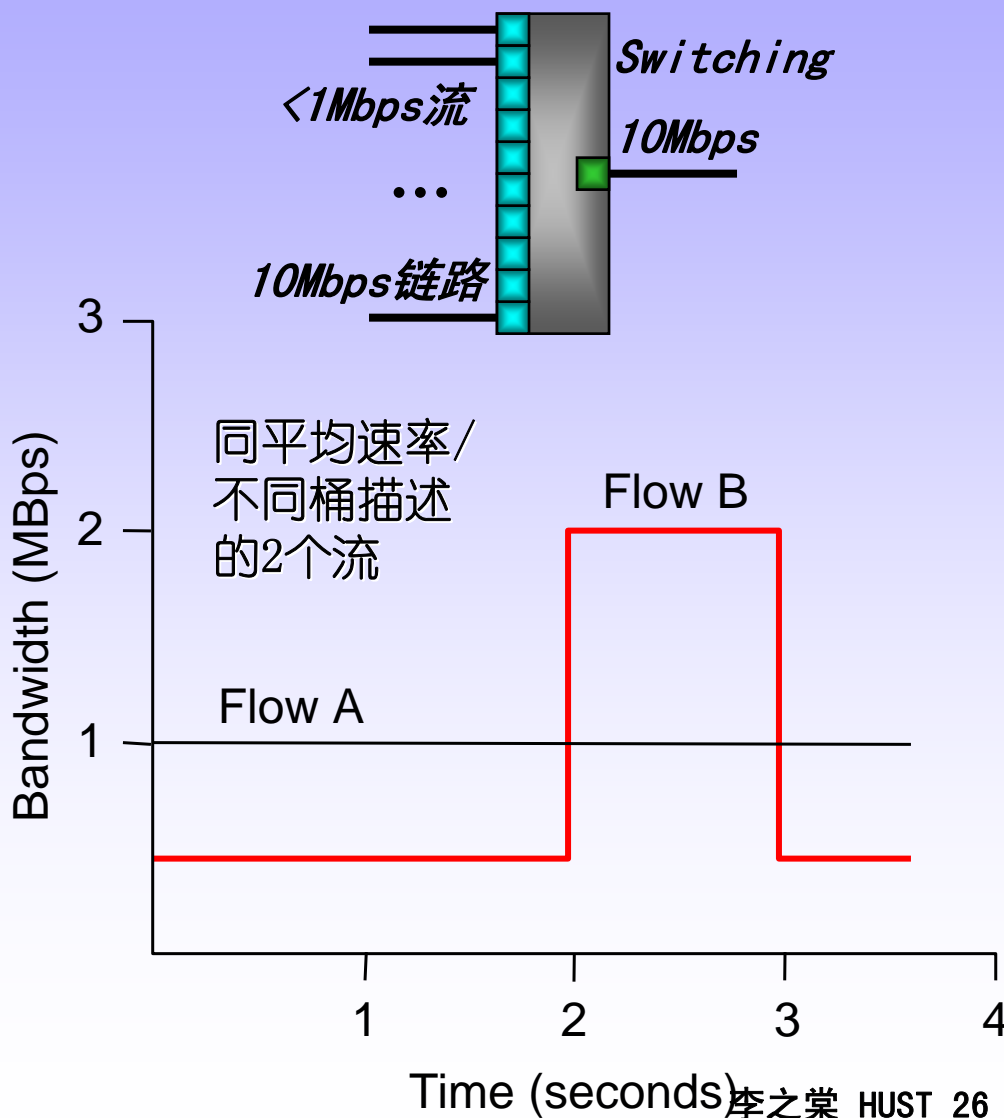
◆ 可将流说明分为2部分

- ♣ 描述通信量特征部分Tspec: 比较复杂
- ♣ 描述网络要求服务部分Rspec容易描述

- ◆ Tspec要给网络提供关于流所使用带宽的足够信息, 以便作出智能的准入决定, 然带宽是变化的: 如一视频在场景变化时比静止时产生更多比特率, **压缩视频**的可变比特率
- ◆ 源端队列管理, 控制延迟和丢包率, 描述源端带宽随时间变化的方法: **令牌桶过滤器: 2个参数:** 令牌速率 r 和桶深度 B (**存储不能及时发送的令牌**), **发送 n 个字节需 n 个令牌开始没令牌, 然后以每秒 r 的速度累计, 可累计不 超过 B 个令牌**
- ◆ 即以尽可能快速度连续向网络发送 B 个字节, 但经过足够长时间后, 每秒发送字节数不能超过 r 个

管理队列/控制延迟/避免丢包

- ◆ **令牌桶过滤器 (Token bucket filter: 令牌速率 r 和桶深 B)** 发送 n 字节需要 n 个令牌
- ◆ 假设每个流按单字节而不是包来发送
- ◆ 流A以1Mbps稳定速率产生数据, 故可用 $r=1\text{Mbps}$, $B=1$ 字节, 即以1Mbps速率接收令牌但存储的令牌不多于1
- ◆ 流B以平均1Mbps速率发送, 但先以0.5Mbps速率发送2秒, 后以2Mbps速率发送1秒方式进行. 故也用 $r=1\text{Mbps}$, 然它需桶深 $B=1\text{M}$ 字节, 以便存储1-2秒间小于1Mbps速率时留下的令牌
- ◆ 1-2秒间, 用1Mbps接收令牌但只以0.5Mbps消耗, 存储 $2 \times 0.5 = 1\text{MB}$, 2-3秒多消耗, 3-4秒又开始存储



(4) 准入控制

- ◆ 假定当前资源可用, 当某新流要接收一特殊级别服务时, 准入控制要:
 - ♣ 查看该流的Tspec和Rspec
 - ♣ 确定能否提供所求服务那么大流量
 - ♣ 并不破坏对前面流的承诺
 - ♣ 难点是何时准入
- ◆ 准入控制依赖请求服务的类型何R中的排队规则
- ◆ 不要把准入控制和策略(Policing) 混淆
 - ♣ 前者是对是否接收每个新流作出决定
 - ♣ 后者是应用于每个包上的动作, 以确保一个流符合为它预留的Tspec. 如某流超出预定2倍速发送, 则丢或打标

(5) 预留协议

- ◆ 如果要从因特网这样的无连接网络上获得实时服务，需要给网络提供更多的信息
- ◆ RSVP是诸多协议中倍受关注的一个，它与传统面向连接网络的信令协议截然不同
- ◆ RSVP一关键假设：它不应该损害无连接网络中存在的健壮性（端到端连通性基本不与网络R的崩溃、重启、链路断开等状态无关），**软状态**维持健壮性
- ◆ RSVP旨在支持多点广播流，使其同单播一样有效
 - ♣ 采用**面向接收方的资源预定**。如1讲多听的Mbone中，有的收方只听一个，有的则要听多个。收比发的应用多
 - ♣ 面向连接中是**面向发送方的预定**，如主叫决定电话信道

好的特性

◆RSVP的软状态与面向收方的性质带来好的特性:

- ♣非常直接增加/减少提供给收方的资源分配级别，这是由于每个收方**定期发送刷新**报文以正确**设置软状态**，使请求新级别资源的新预定变得很容易
- ♣万一主机崩溃，主机分配给一个流的资源会自然超时并被释放

◆How资源预定？R或链路失效会发生些什么？

发收双方间怎样获得预定

- ◆ 第1：收方要知道发方可能发送的Traffic以便进行预定，即要知发方的Tspec
- ◆ 第2：要知包从发方到收方遵循什么路径，以便能在路径的每个R上建立一资源预定
- ◆ 可由发方向收方发一含Tspec的报文来满足上述两要求，结果
 - ♣ 收方得到了Tspec
 - ♣ 途中每个R查看该报文（称PATH报文）并算出反向路径，该路径将被收方用来向发方返送“路径中每个R有关的预定”

资源预定的建立

◆ 收方收到PATH报文后，用RESV报文沿多播树向上发回一个预留

- ♣ RESV包含发方的Tspec和描述收方请求的Rspec
- ♣ 路径上的每个R**查看预定**请求并**分配必要资源**来满足
 - ☞ 如果该预定能做，则RESV被传给下一R
 - ☞ 如果不能则向发出请求的收方返回一个出错报文
- ♣ 如果一切正常，路径中的每个R上都安装了正确的预定，只要接收方想保留该预定，它就每30s再发一次相同的RESV报文

R和链路失效时？

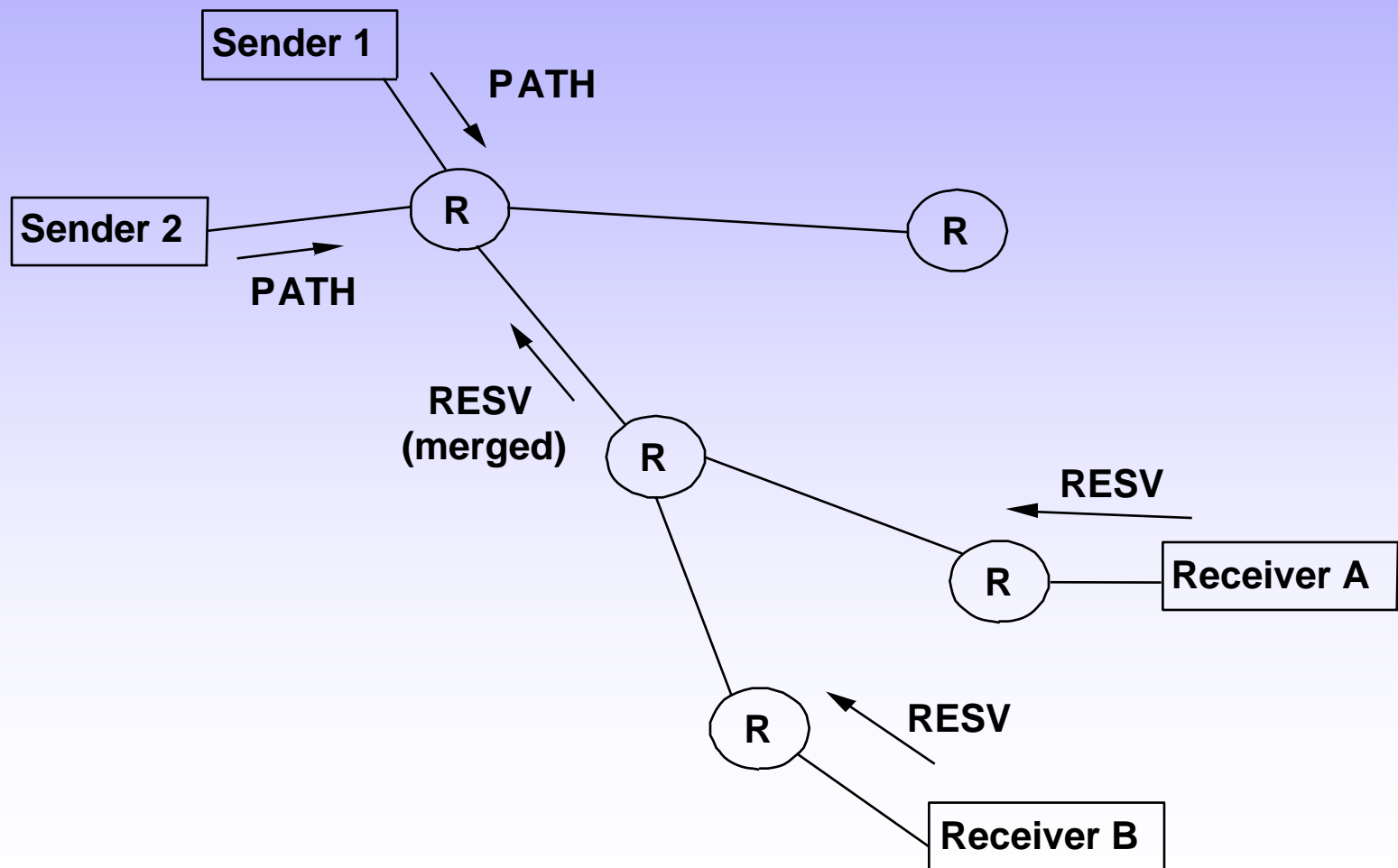
- ◆ 当链路或R失效时，路由协议将生成一条从发方到收方的新链路
- ◆ PATH报文每30秒发一次，若R在其转发表中检测到变化发生，还可能更快发送一Path
- ◆ 新路由稳定后的第一个Path报文将通过新路径到达收方
- ◆ 收方的第一个RESV将遵循新的路径并在新路径上建立新的预定
- ◆ 不在新路径上的R将停止接收RESV报文，它们的预定将超时并被释放

多播树上进行预定

◆ 多个收方对一个发方

- ♣ 当一RESV沿多播树向上传时，很可能碰到该树的某部分有某些收方已建立的预定
- ♣ 若某点上预定的资源足以满足这2个收方的预定，如收方A已有 $\leq 100\text{ms}$ 延迟预定，收方B要求 $\leq 200\text{ms}$ ，则B不需再预定
- ♣ 若新请求是 $\leq 50\text{ms}$ 延迟，那么R先要看看能否接受该请求，若能则向上发请求。若收方A下一次再请求 $\leq 100\text{ms}$ ，R就不在传递该请求

多播树上进行预定



多个收方对多个发方

- ◆ 收方需收集所有发方的 T_{spec} ，并足够**预留以容纳**所有发方的通信量
- ◆ 并不要把所有 T_{spec} 累加，如10人音频会议并不一定分配足够资源来传送10个音频流，因2个讲话者预定可能足够
- ◆ 须根据具体情况由发方的 T_{spec} 计算出总的 T_{spec} ，可能只愿听讲话人中的一部分人的讲话
- ◆ RSVP有不同的预定风格
 - ♣ 为所有讲话者预定资源
 - ♣ 为任意 n 个讲话者预定资源
 - ♣ 只为讲话者A和B预定资源

包的分类与调度

- ◆ 完成通信量、所需服务及合适预定的描述后，R的工作就是向包实际提供所需服务，服务分为2类
 - ♣ 将每个包与适当的预定相关联以便得到正确处理，这一进程称为**分类**classifying
 - ♣ 管理队列中的包以便它们能接收到所请求的服务，这一进程称为**调度**scheduling

包的分类

- ◆ 检查包中最多5个字段：源地址、目地址、协议号、源端口、目端口
- ◆ 根据这些信息将包归为适当的类，如分类为受控负载，或保障流（1对1影射），完成从包头中流说明到包类ID的影射
- ◆ FIFO排队机制不适宜提供许多不同层次的服务。加权排队机制可提供保障服务

可扩展性问题

- ◆ 尽管集成服务和RSVP能提高IP best-effort，但服务提供商还是感到不是好的模式，原因是可扩展性是IP的基本设计目标
- ◆ best-effort模式下，对每个流很少存其信息或是无状态，随网络的增长，R要使每个流要更快，处理大量路由表
- ◆ RSVP则要为每个流预留带宽

面向流的预留

- ◆ 设在OC-48 (2.5 Gbps) 链路上, 一个流表示64Kbps视频流
 - ♣ 则有 $2.5 \times 10^9 / 64 \times 10^3 = 39000$
 - ♣ 每个流还需存储相应状态并周期刷新
 - ♣ R对它们分类、控制和排队
 - ♣ 准入控制每次决定对每个流资源预留
- ◆ 需反推机制, 防止用户长时间制造很大的预留

2.9.3 区分服务

- ◆ 综合服务的主要思想是：给**单独的流分配资源**
- ◆ 区分服务的主要思想是：给几类（实际上是2类）**通信量分配资源**
- ◆ RSVP是通过协议来告诉所有路由器某个流是高级包
- ◆ DiffServ是让这些**包到达时告诉路由器**自己的重要程度

区分服务的两个问题

◆ 谁来设置包的类型，在什么环境下设置？

- ♣ 常用办法是在管理**边界设置**这一点，如ISP的边界网关，因为其接入用户可能已付比best effort更高的费用，可配置包高至**某一最大速率**，其他包保持尽力服务

◆ 当R看到这些有标记的包时，做哪些不同的处理？

- ♣ IETF正在标准化R对标记包的行为，重新定义TOS→6位分配成区分服务代码**指针DSCP**：**每跳行为PHBs**，包头中不止1位来表明多个R的行为

加速转发EF-最简单的PHB

- ◆ 对标记为EF的包，R以最少延迟和丢失率转发该包
- ◆ 使R能对所有EF包都**遵守这一承诺**的唯一办法是，严格限制EF包到达**R的速率小于该R转发该EF包的速率**（即 $I_r < O_r$ ），还要给R留有**偶尔发送其他包的资源**
- ◆ 保守但简单的办法是：进入该域的所有EF包的**速率总和应低于**该域中最慢链路的带宽

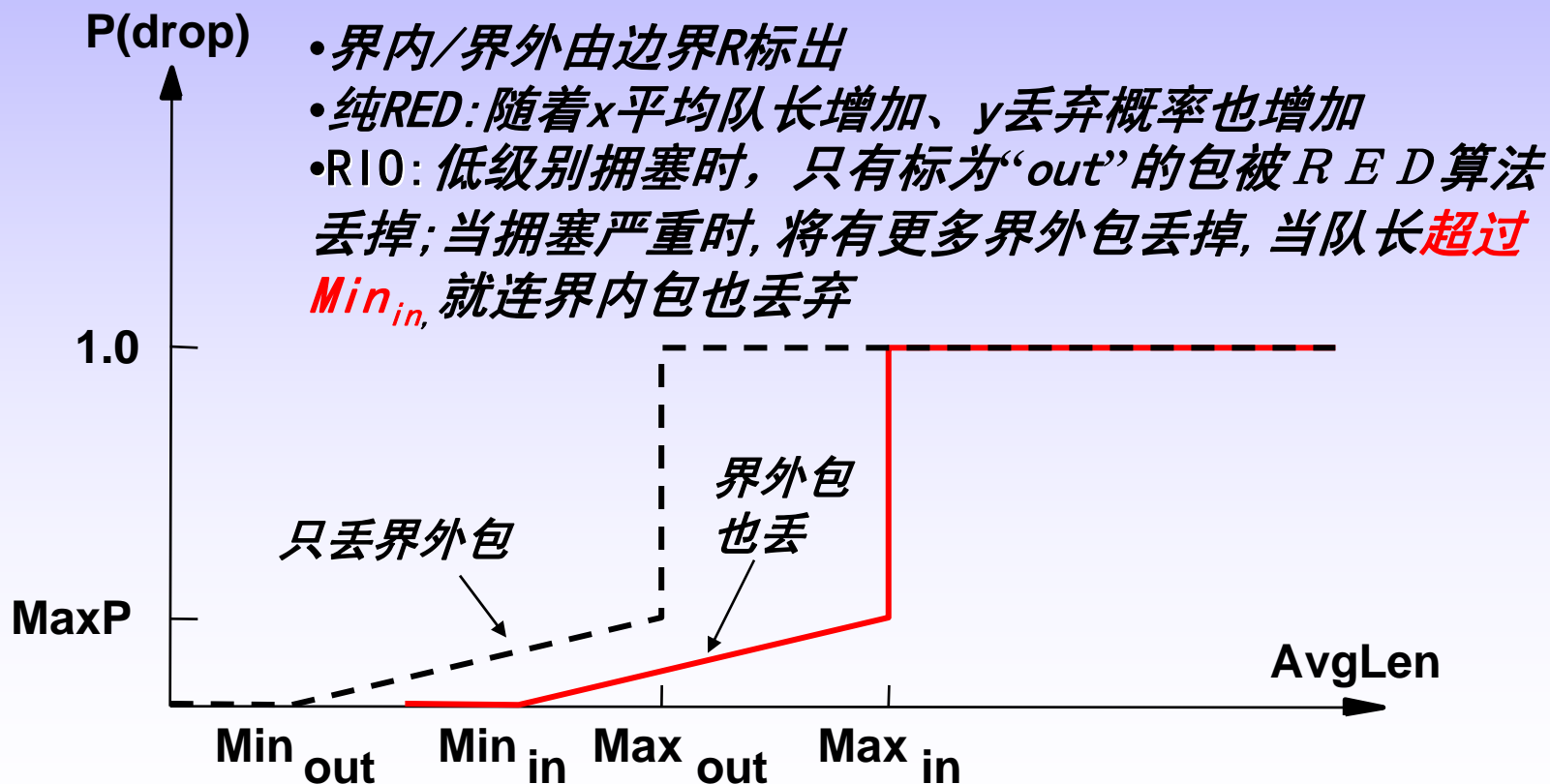
实现EF包的2种策略

- ◆ 给EF包赋予**比其他包优先**的权利
- ◆ 在EF包和其他包之间执行加权平均排队，将EF包的权值设置成足够大，使其可快速转送。这一策略较因即使在EF通信量很大情况下，仍可确保非EF包获得一定的链路速率
- ◆ 这意味着**不能总是确保对EF包的服务**，但却可**防止过载时其他通信量被锁在网络外面**

确保转发AF-另一PHB

◆ 有2种对RED改进的方法: **加权RED**

♣ 带**界内/界外**2个丢弃概率的 RED (RIO)



界内界外的标识

- ◆ 管理域上的边界路由器对包打上界内界外的标记. 如ISP和客户达成配置是:
 - ♣ 允许客户X最多发送yMbps的可确保的通信量
 - ♣ 只要X低于yMbps, 其所有包将被标记为**界内**
 - ♣ 一旦超过了该速率, 额外的包将被标记为**界外**
- ◆ RIO的也很依赖参数的选择, 目前工作?
- ◆ RIO不改变界内外包的顺序, 这对TCP很重要
- ◆ 加权RED: 提供2个以上的丢弃概率曲线, DSCP的值用于选择其中哪条曲线

区分服务的第3种方法

- ◆ 用DSCP的值决定将包放在哪个队列, 队列由2.2.2节加权排队算法管理. 如
 - ♣ 用一个DSCP指向“尽力服务队列”, 第二个DSCP指向“高级”队列
 - ♣ 为高级队列选一个加权值, 如令尽力队列是4, 高级队列是1, 使其获得比尽力服务包更好的服务, 则**确保高级包获得的带宽**为
 - ♣ $B_{\text{premium}} = W_{\text{premium}} / (W_{\text{premium}} + W_{\text{best_effort}}) = 1 / (1 + 4) = 0.2$
 - ♣ 即为高级包预定了链路的20%



第6章习题

◆ 2;4;6;8;13;16;20;25;27;32;42;45



Thank
you!



Questions