



计算机网络应用层： QoS概述

刘志敏

liuzm@pku.edu.cn



相关知识回顾

■ 数字音频与数字视频的特点

- 数据量大，采用数据压缩技术，
- 实时性：RTP首部含有顺序、时间戳、编码等信息

■ 媒体播放器与媒体服务器间的协议

- RTCP：交换数据传输质量，如丢帧率、延迟等信息
- RTSP：传输的控制协议、用户界面、解压缩、消除差错、缓存及播放

■ 媒体播放器

- 支持RTP、RTCP、RTSP
- 功能：用户界面、解压缩、消除差错、缓存及播放

■ SIP：VoIP信令协议，支持移动性

■ 如何保证传输质量，改善用户体验？



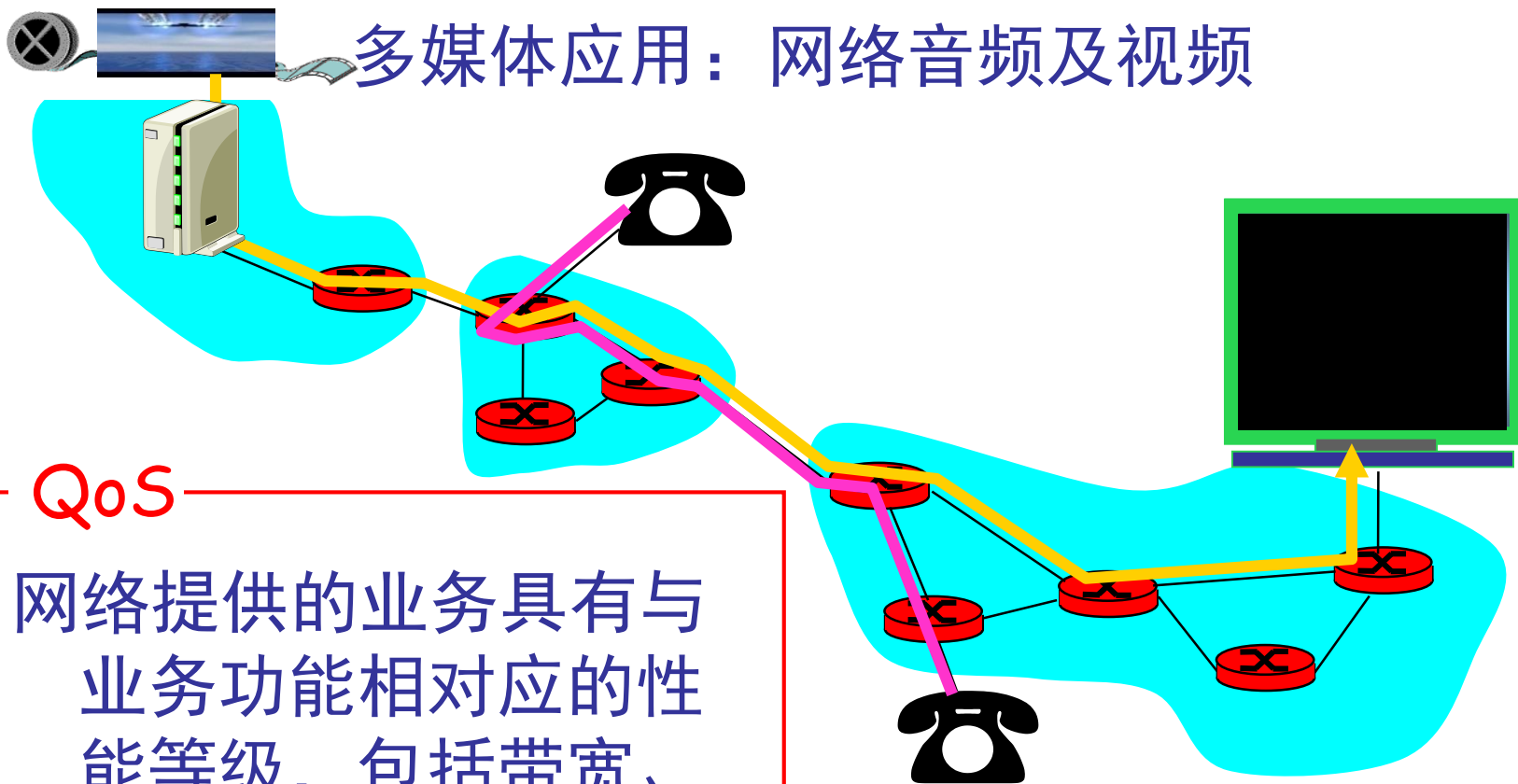
QoS概述

- 理解QoS
- QoS机制：
 - RSVP
 - DiffSer模型
 - 调度机制与漏桶算法
- 其他网络技术
 - Web 与 HTTP
 - CDN：内容分发网络
 - MPLS：多协议标记交换

理解QoS (Quality of Service)

数字媒体：利用网络提供信息和娱乐服务

多媒体应用：网络音频及视频



QoS

网络提供的业务具有与业务功能相对应的性能等级，包括带宽、延迟、抖动、丢帧率

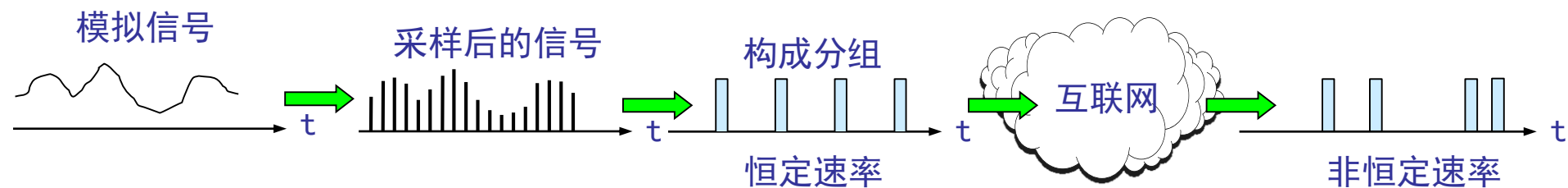
多媒体业务的服务质量

QoS (Quality-of-Service) : 满足业务需求的技术指标, 如带宽、延迟、延迟的变化、丢帧率

QCI	Resource Type	Priority	Packet Delay Budget	Packet Loss Rate	Example Services
1	GBR	2	100ms	10^{-2}	Conversational Voice
2		4	150ms	10^{-3}	Conversational Video (live streaming)
3		5	300ms	10^{-6}	Non-conversational Video (buffered streaming)
4		3	50ms	10^{-3}	Real Time Gaming
5	Non-GBR	1	100ms	10^{-6}	IMS Signalling
6		7	100ms	10^{-3}	Voice, Video (Live Streaming), Interactive Gaming
7		6	300ms	10^{-6}	Video (Buffered Streaming); TCP-based (e.g. www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
8		8			
9		9			

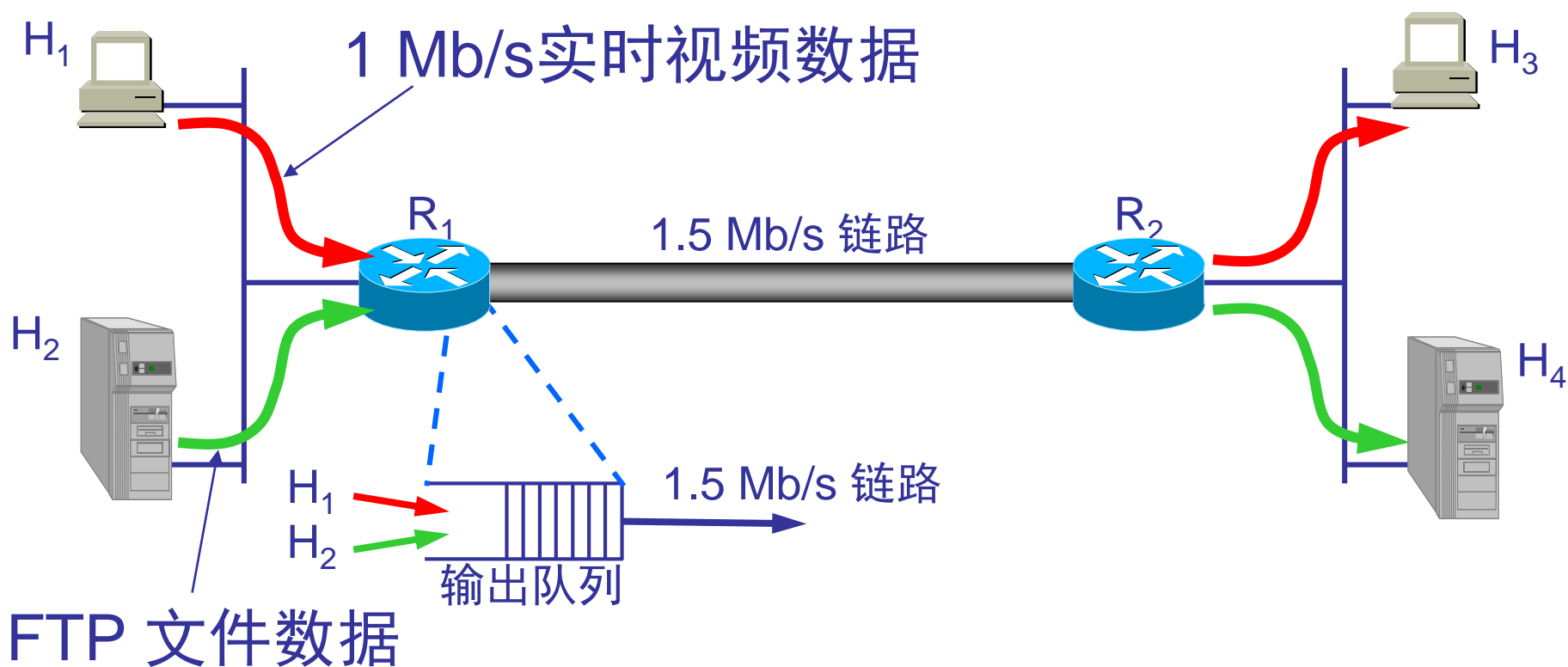
互联网提供尽力而为服务

- 模拟信号经采样、量化及编码转为数字信号，再封装为分组
- 发送的媒体数据报文（RTP）是等间隔的，经过互联网存储转发后为不等间隔的
- 时延抖动(Jitter)：由于网络延时的变化导致分组到达速率的变化。如果对存在着严重时延抖动的网络不进行处理，就会使人感觉话音忽快忽慢，用户难以接受。



数据流经过存储转发，速率及延迟发生变化

主机H1与H3之间传输的视频信息，需要保证实时性
主机H2和H4之间传输文件，需要保证可靠性





QoS概述

- 理解QoS
- QoS机制：
 - RSVP：综合服务
 - DiffSer模型
 - 调度机制与漏桶算法
- 其他网络技术
 - CDN：内容分发网络
 - MPLS：多协议标记交换



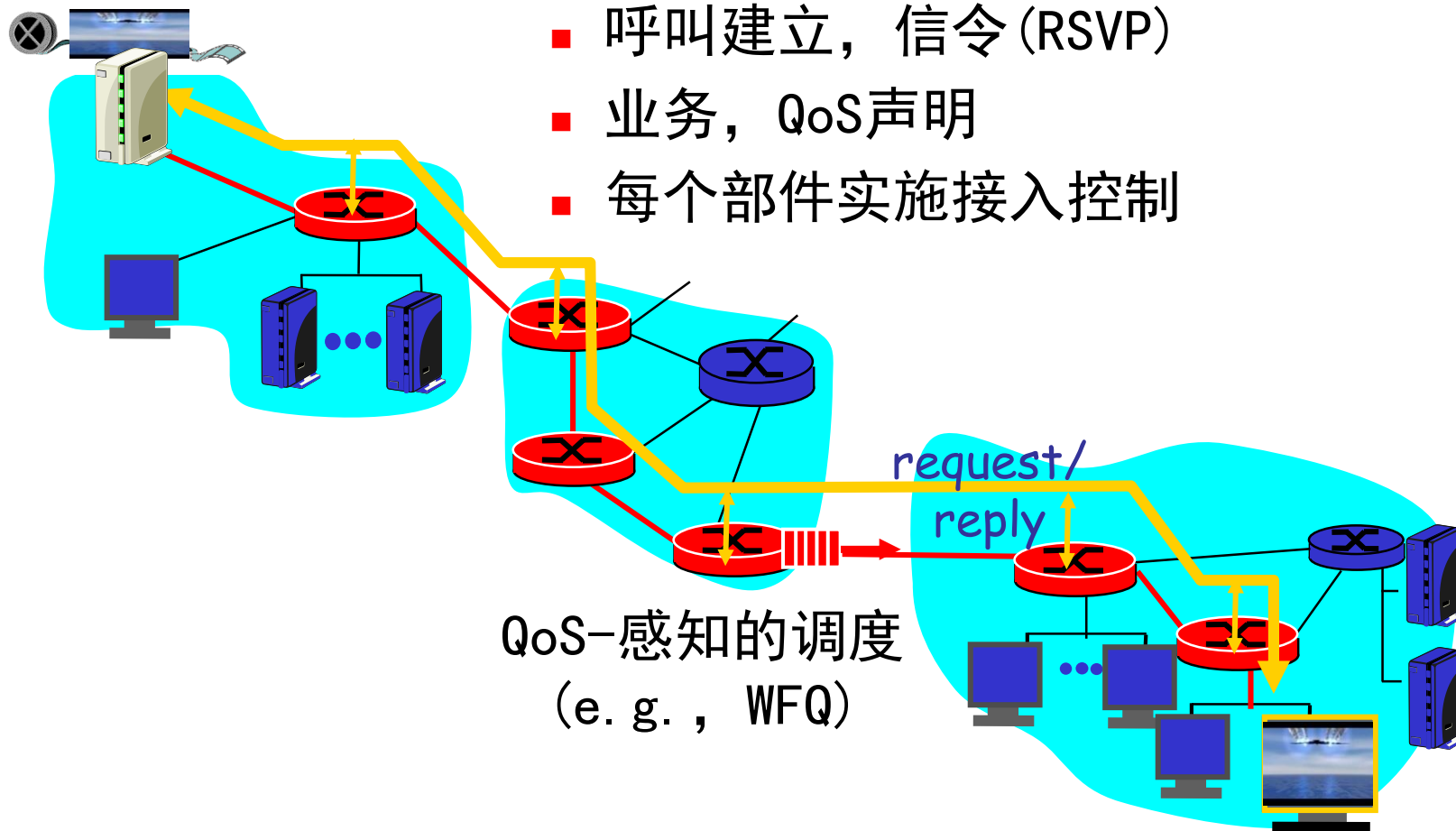
综合服务与资源预留

- IntServ (Integrated Services)对每个应用会话提供服务质量保证，其特点**资源预留**和**呼叫建立**
- **资源预留**：路由器需要知道为某一会话预留了多少资源（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。

保证QoS的场景

■ 资源预约

- 呼叫建立，信令 (RSVP)
- 业务，QoS声明
- 每个部件实施接入控制





接纳控制 Call Admission

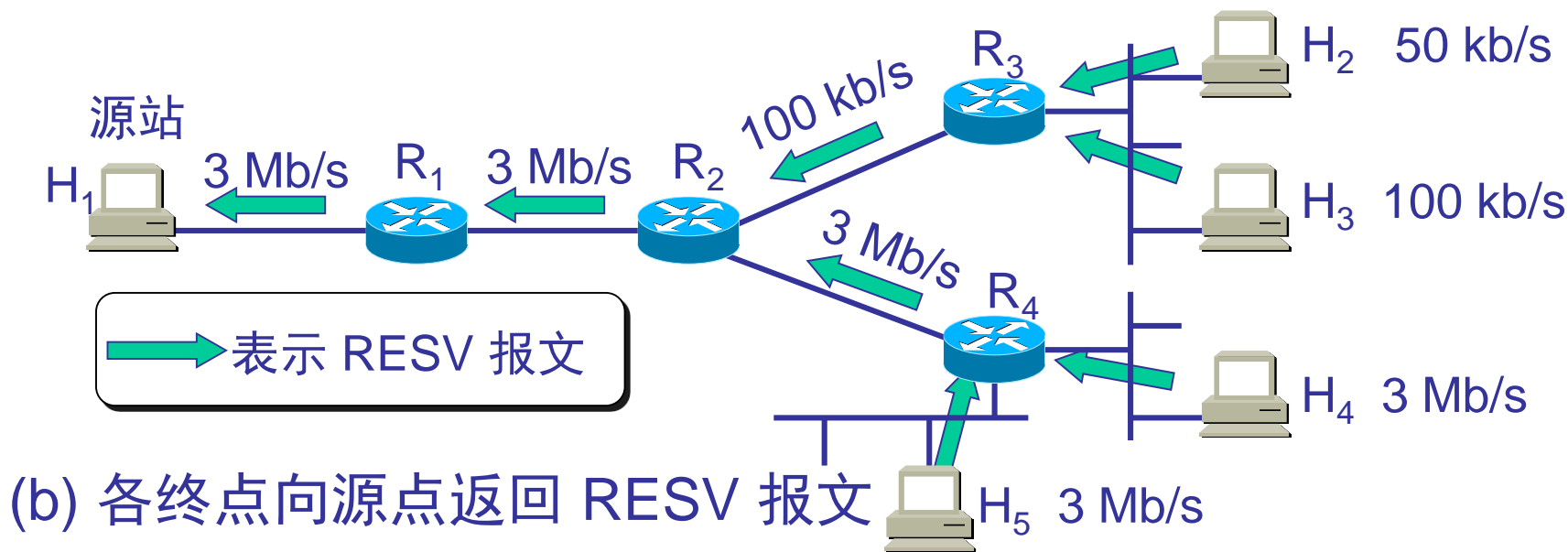
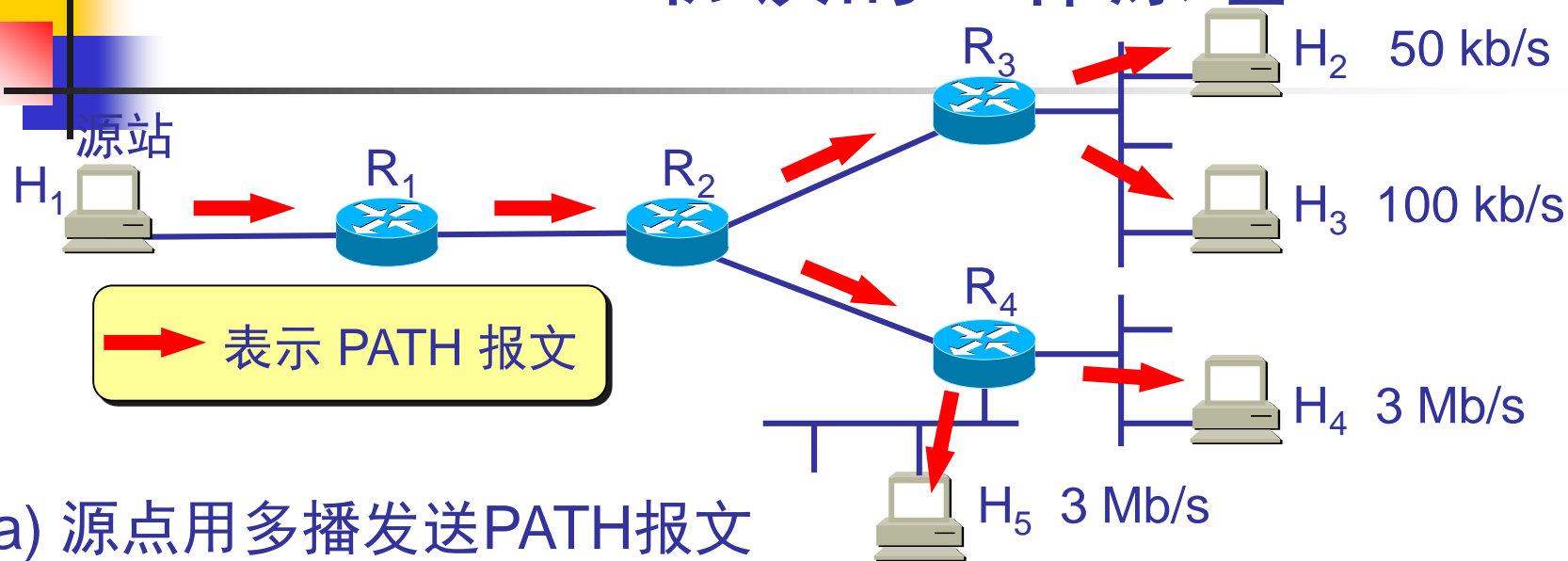
- 资源预约：路由器维护状态信息、分配资源
 - 允许或拒绝建立一个新呼叫
- 到达的会话必须：
- 声明其QoS需求
 - **R-spec**：定义需要的QoS
 - 网络传输的业务流特征
 - **T-spec**：定义业务流特征
 - 信令协议：携载R-spec及T-spec给路由器，以预约请求
 - **RSVP** [RFC 2205]



RSVP：基本过程

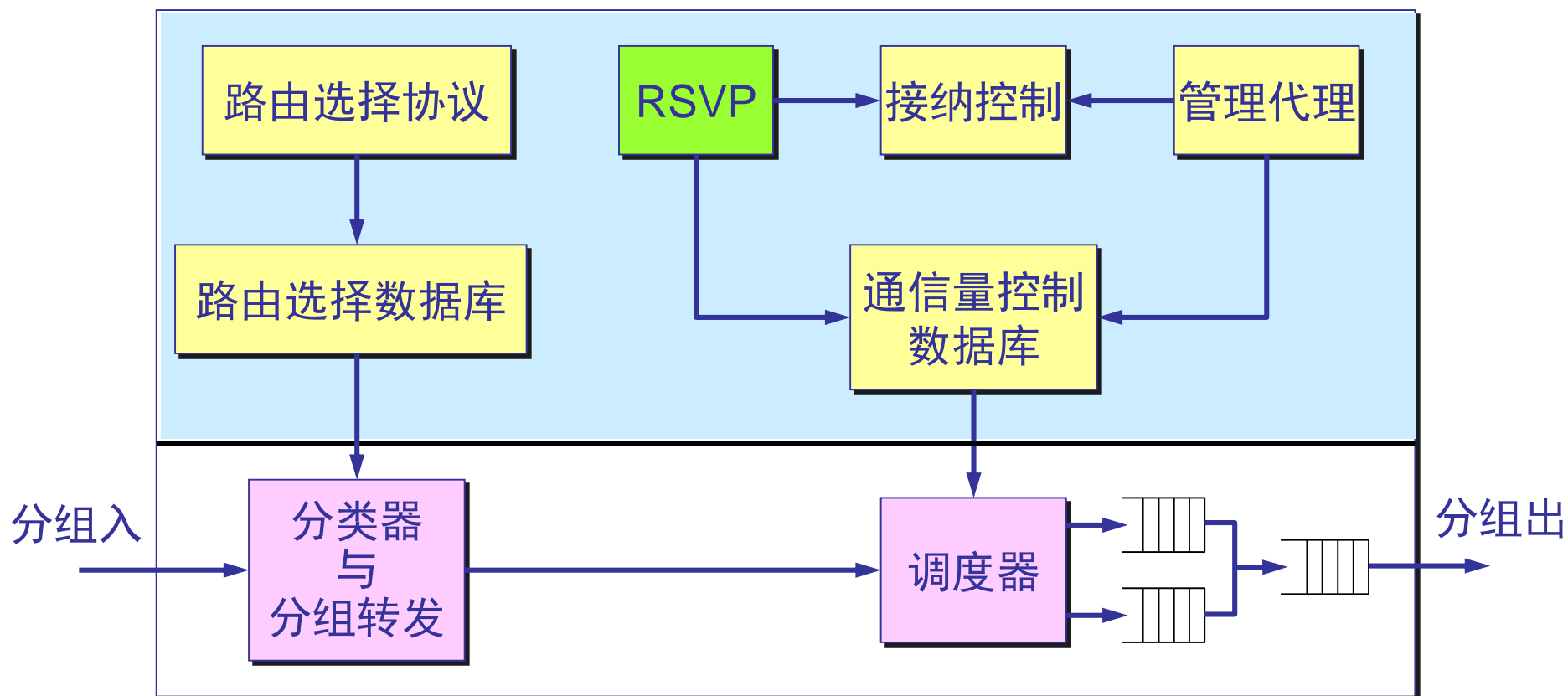
- 发端、收端加入一个多播组
 - 在RSVP之外完成
 - 发端不需要加入组
- 发端到网络的信令
 - 路径信息：保证发端知道所有的路由器
 - 路由拆除：从路由器上删除发端路径状态信息
- 收端到网络的信令
 - 预约信息：预约由发端到收端的资源
 - 预约拆除：删除接收端预约
- 网络到端系统的信令
 - 路径错误
 - 预约错误

RSVP 协议的工作原理



IntServ 在路由器中的实现

IntServ组成：RSVP、接纳控制、分类器、调度器





IntServ体系结构存在的问题

- (1) 状态信息的数量与流的数量成正比。在大型网络中，按每个流预留资源，开销大
- (2) IntServ 体系结构复杂。若要保证服务质量，所有的路由器都必须配备RSVP、接纳控制、分类器和调度器



QoS概述

- 理解QoS
- QoS机制：
 - RSVP
 - DiffSer：区分服务
 - 调度机制与漏桶算法
- 其他网络技术
 - CDN：内容分发网络
 - MPLS：多协议标记交换

提供划分等级的服务

■ 提供比尽力而为更好的服务

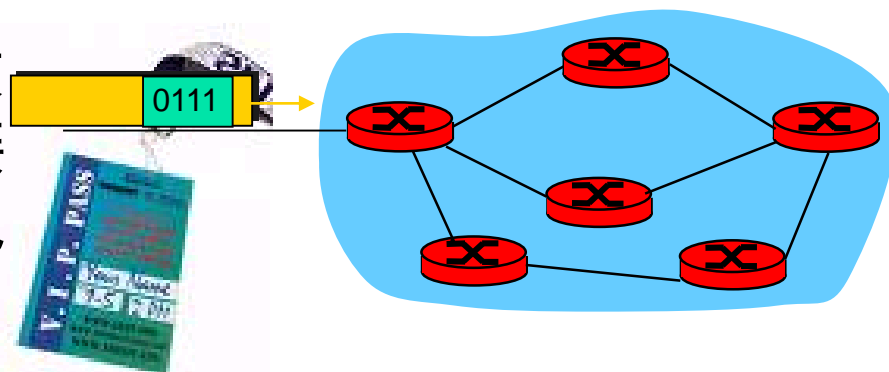
- 一种模型适于所有业务模型

■ 另一种选择：提供多等级的服务

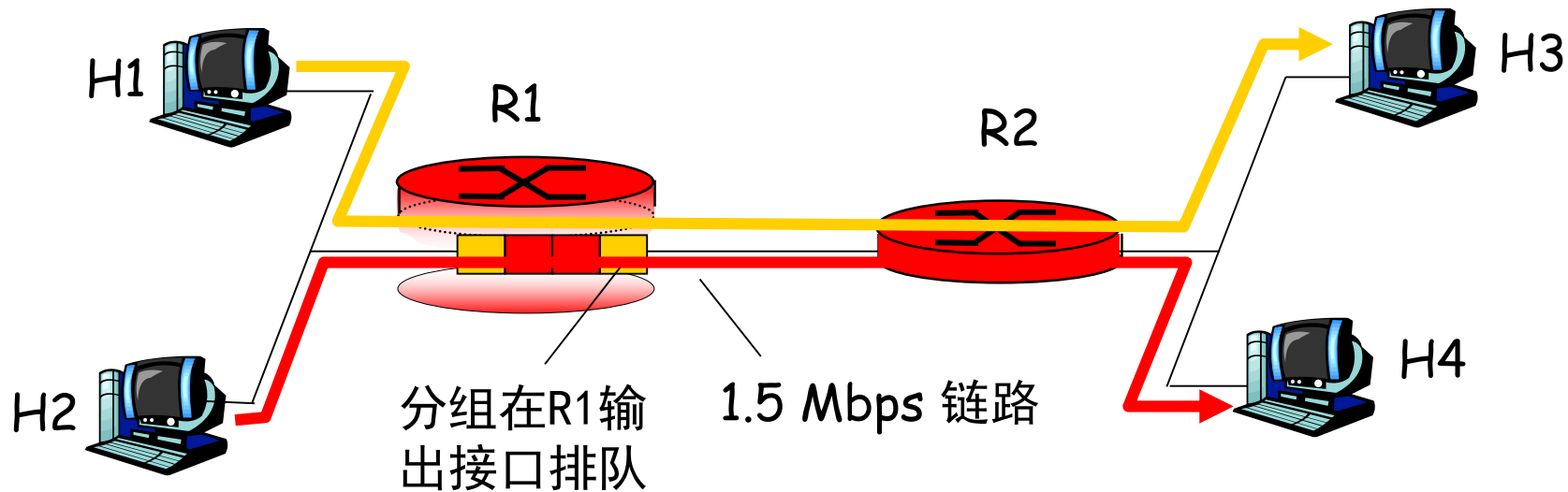
- 对服务类型划分等级
- 等级不同，网络处理方式也不同（比如：区分VoIP业务与一般业务）

■ 聚合流：不同业务等级的数据流，可能来自相同的连接，不能用（IP1，IP2）标记服务类型

■ 利用IP分组头中ToS字段（服务类型）标识业务类型

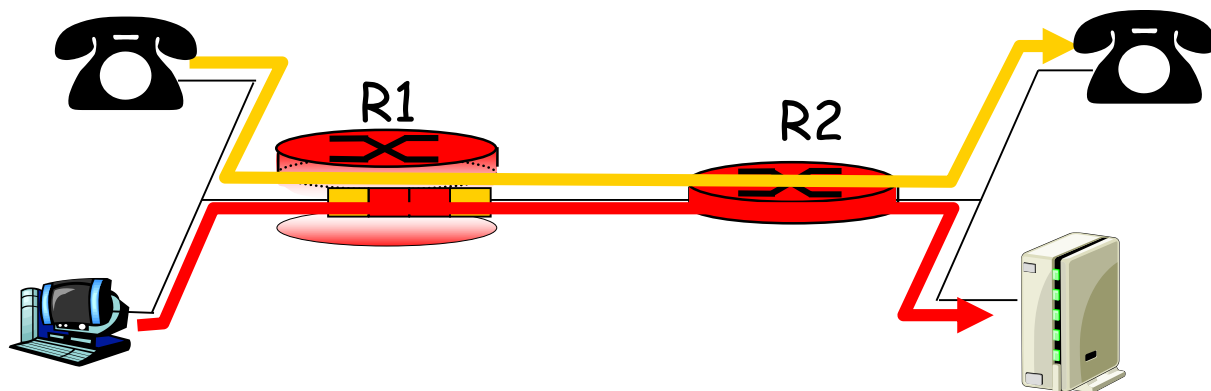


多等级服务的场景



场景 1：一个FTP流与一个音频流

- 例如：1Mbps的IP电话与FTP共享1.5Mbps链路
 - FTP突发数据可能阻塞路由器，导致语音分组丢失
 - 需要语音分组传输优先级比FTP的更高

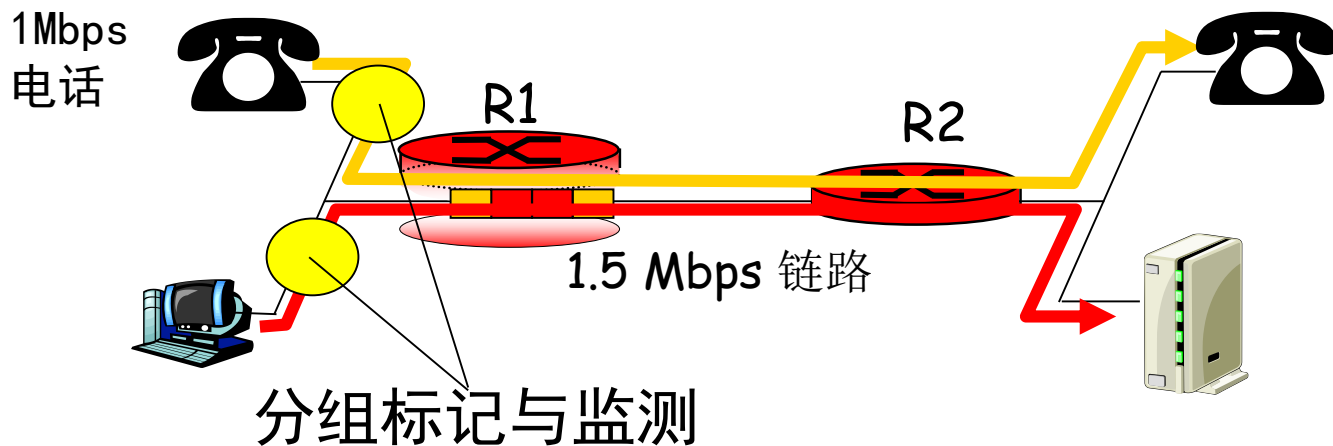


方案 1

标记分组，使得路由器可以区分不同类型的分组流；需要路由器提供处理分组的新策略

保证QoS的原理：（续）

- 若应用程序的行为不当，情况会怎样？（语音发送速率高于其声称的速率）
 - 监测：迫使信源限制其占用的带宽资源
- 在网络边界标记并监测

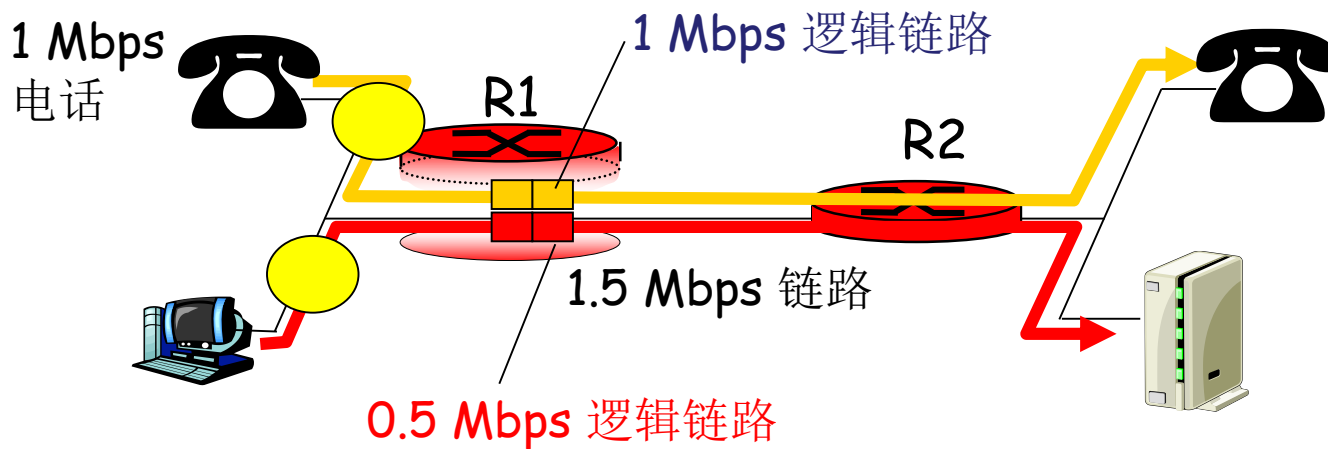


方案2

在各业务类型之间，提供保护与隔离

保证QOS的原理：（续）

- 为业务流分配固定（非共享）带宽：不传输超出分配带宽的业务流



方案 3

在流量类型或流之间提供隔离时，尽可能有效地使用资源



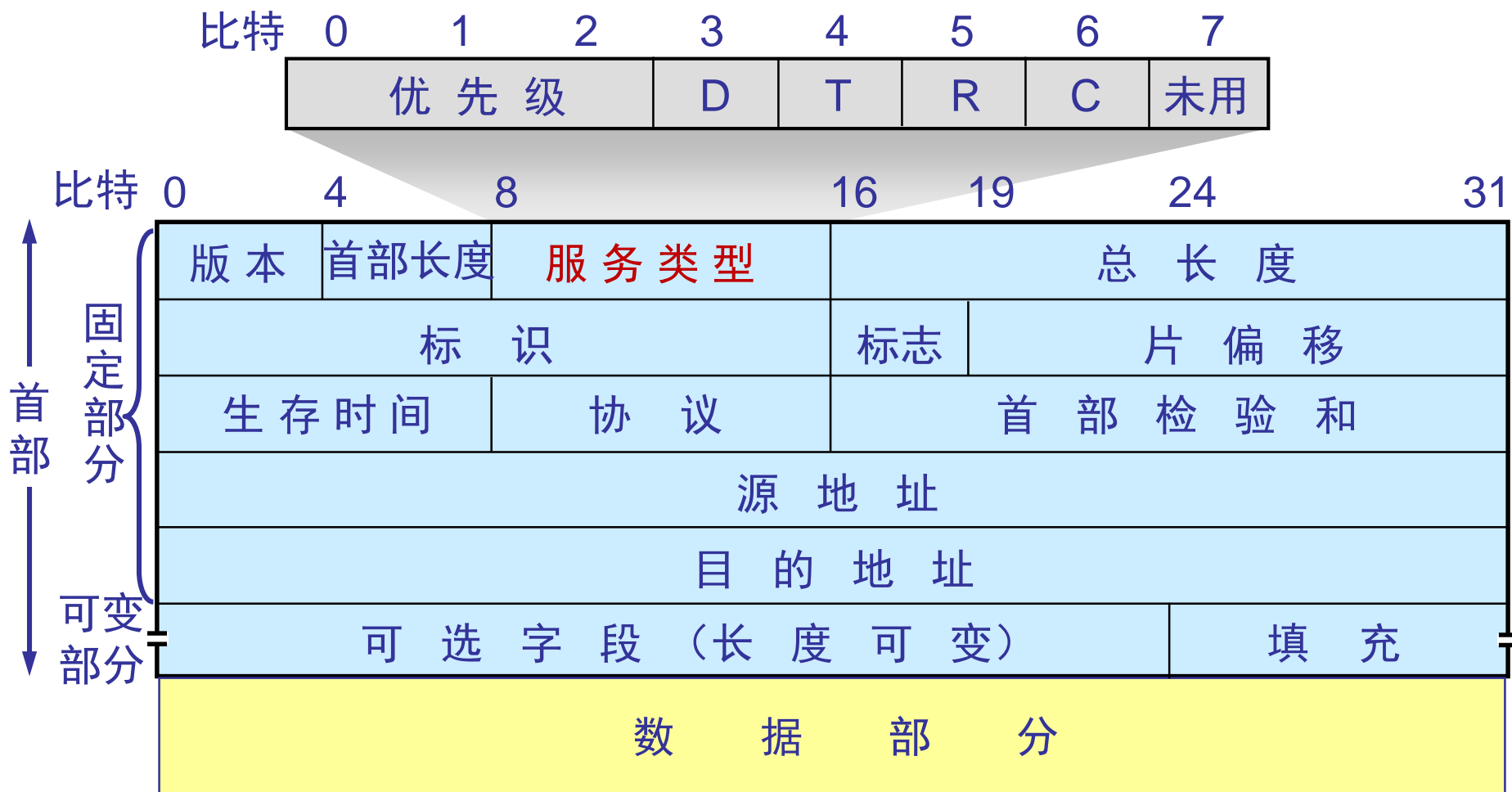
区分服务 DiffServ

区分服务 DiffServ (Differentiated Services)

- (1) 在路由器中增加区分服务的功能
- (2) 将网络划分为许多DS域
- (3) 边界路由器：分类、标记、整形、测量
- (4) 聚合：根据流的DS值将若干个流聚合成更少的流

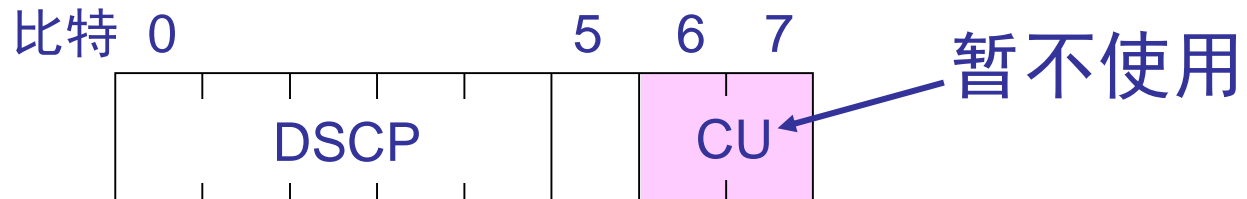
IP首部中的TOS字段

- 由首部和数据两部分组成，首部占 20 字节



区分服务 DiffServ

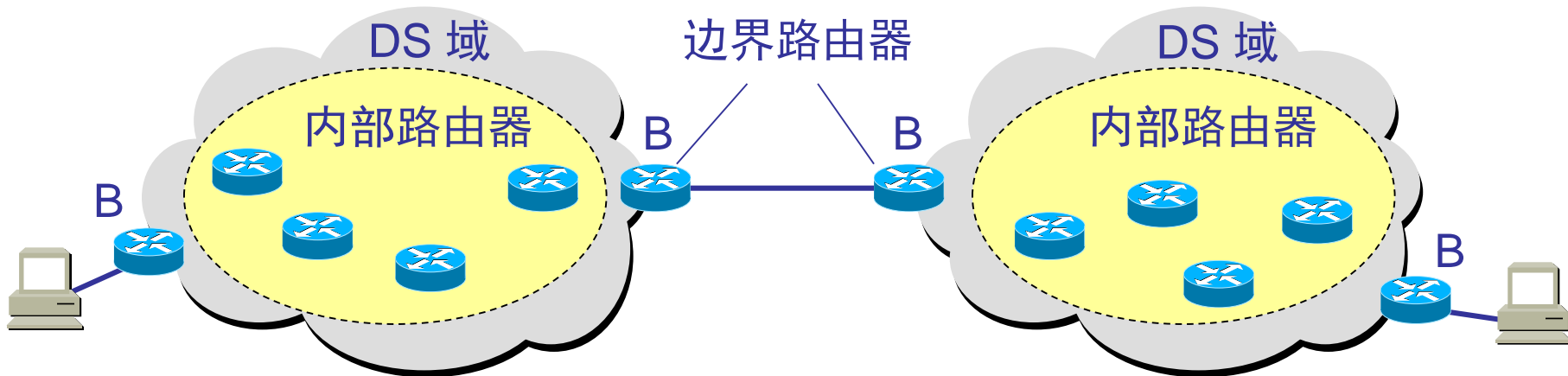
- (1) 在路由器中增加区分服务的功能
- IP头部的“服务类型”定义服务类型
 - 路由器根据 DS转发分组
 - **区分服务码点 DSCP** (Differentiated Services CodePoint): 占6位。
 - 服务等级SLA(Service Level Agreement): ISP 与用户协商**服务等级**, 约定服务类别及业务量
 - 第0~2位表示四级业务
 - 第3~5位表示三种“丢弃优先级”



区分服务 DiffServ

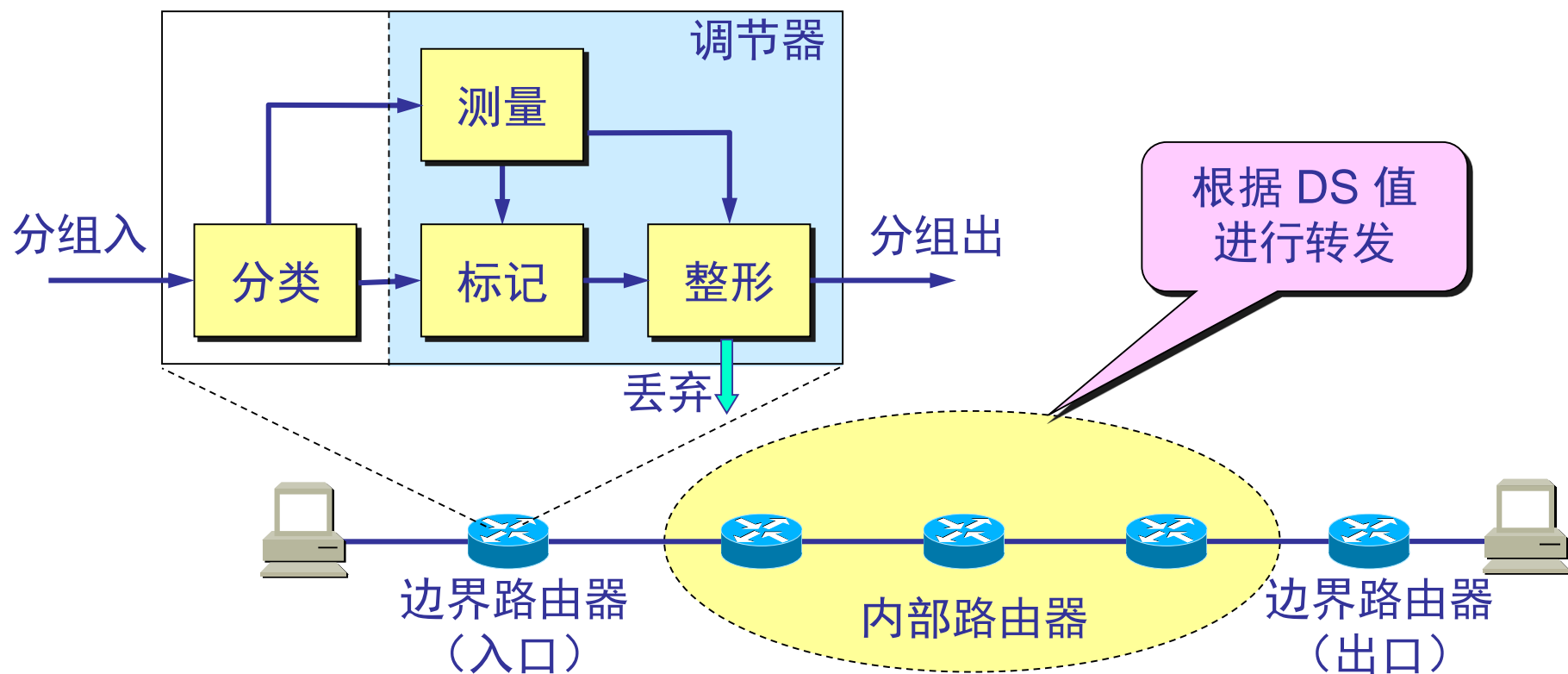
(2) 将网络划分为许多DS域

- 在DS域的边界路由器中实现复杂功能，DS域内路由器尽可能简单，利于快速转发



区分服务 DiffServ

(3) 边界路由器实施分类、标记、整形、测量





区分服务 DiffServ

(4) 聚合(aggregation)

- 根据流的DS值将若干个流聚合成少量的流，无需为每个流维持供转发的状态信息
- 路由器对相同DS值的流按相同的优先级转发，简化内部路由器的转发机制
- 区分服务DiffServ不需要支持RSVP



区分服务 DiffServ

- 路由器如何处理转发的分组？各路由器的行为彼此独立！
- **迅速转发：** 路由器提供大于某一阈值的发送速率，提供保证带宽的端到端服务：低丢失率、低时延及时延抖动
- **确保转发：** 用DSCP表示的四级业务，每级提供最低带宽和缓存空间；对三种“丢弃优先级”，发生拥塞时先丢弃较高“丢弃优先级”的分组

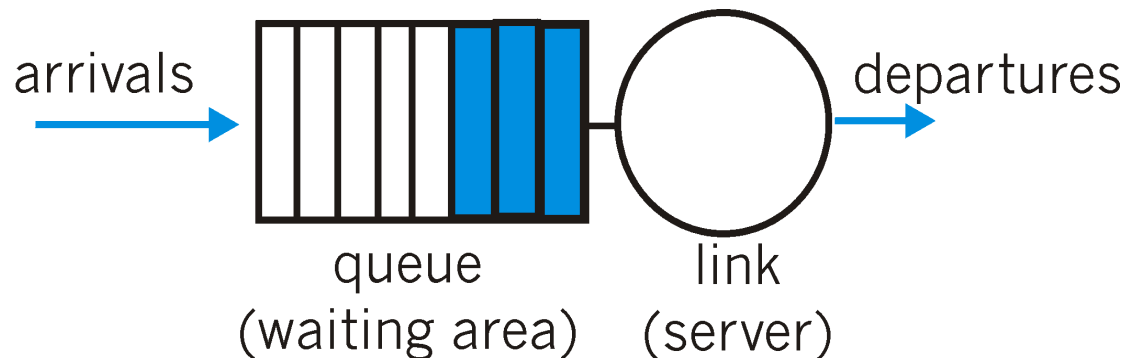


QoS概述

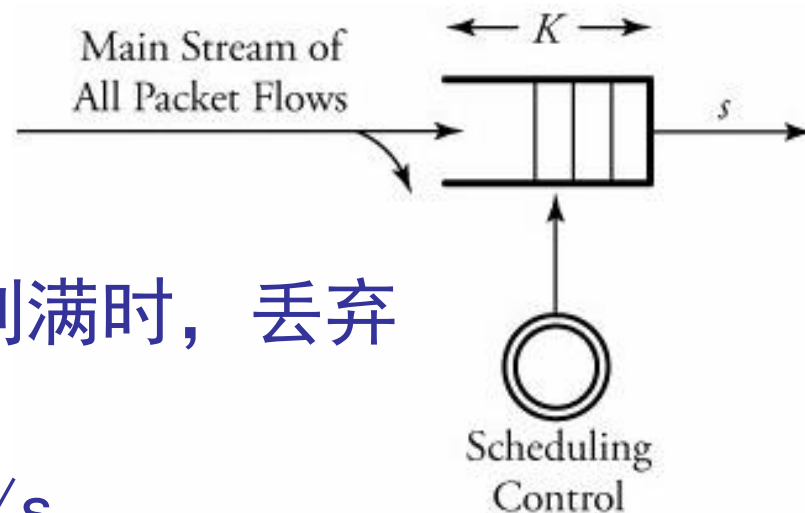
- 理解QoS
- QoS机制：
 - RSVP
 - DiffSer模型
 - 调度机制与漏桶算法
- 其它网络技术
 - Web 与 HTTP
 - CDN：内容分发网络
 - MPLS：多协议标记交换

调度机制

- **调度scheduling**: 选择在链路上要发送的下一个分组
- **FIFO(first in first out)** 先进先出: 按到达队列的顺序发送
 - **丢弃策略**: 若分组到达时队列满了, 丢弃哪个分组?
 - 丢尾: 丢弃新到的分组
 - 优先级: 选择更低优先级的分组丢弃
 - 随机: 随机丢弃



先进先出 FIFO



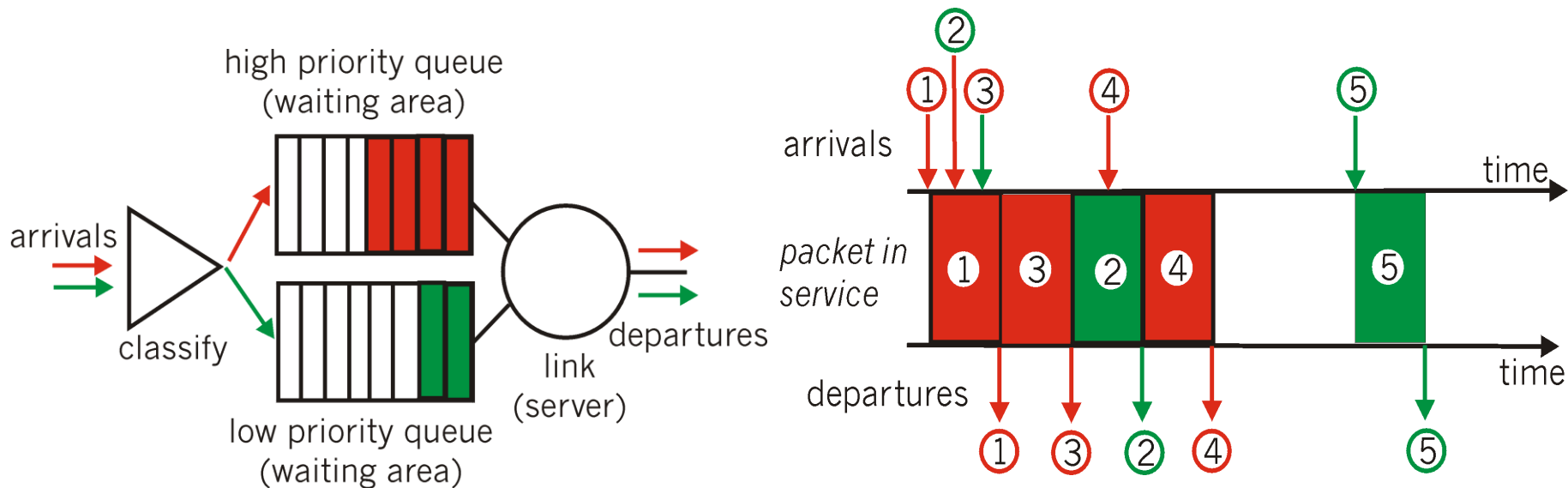
- 先进先出 FIFO，当队列满时，丢弃后新到的分组
- 调度时延的上限： $T_q \leq K/s$ ， K 为缓存深度， s 为输出链路速率
- 缺点：不区分分组，不公平（信源速率高的占用更多的带宽）
- 增加按优先级排队，使高优先级的分组优先得到服务

调度机制: (续)

优先级调度: 优先发送高优先级的分组

- 多种业务类型, 不同的优先级

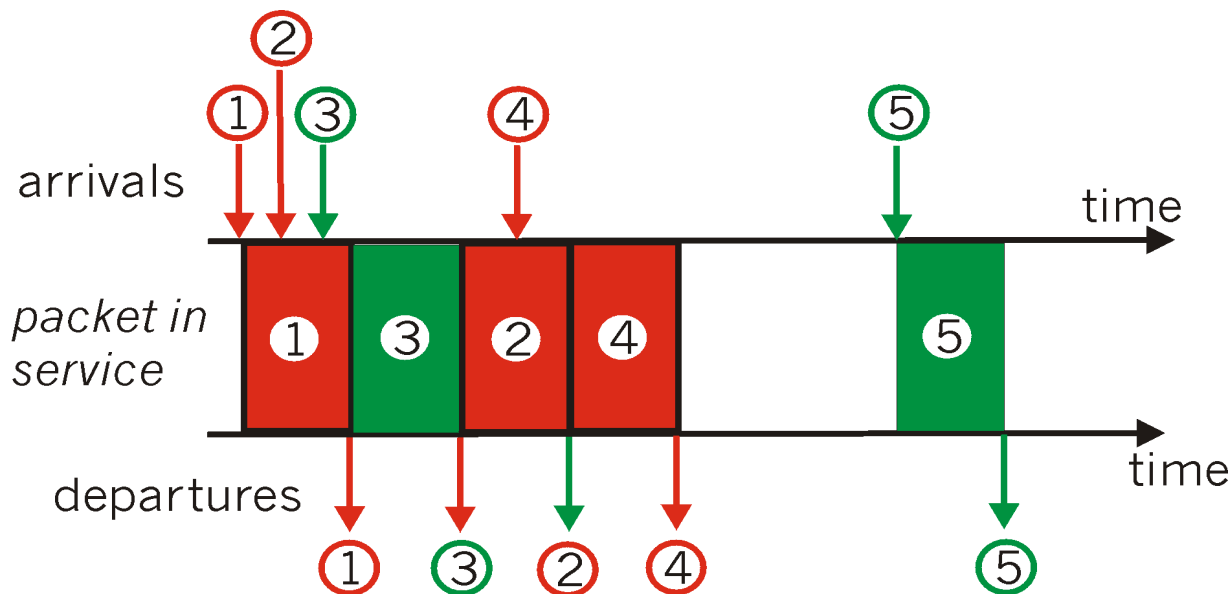
- 分类器: 根据分组或分组头信息进行标记, 如IP源/目的地址, 端口号等



调度机制: (续)

循环调度round robin (最具公平性) :

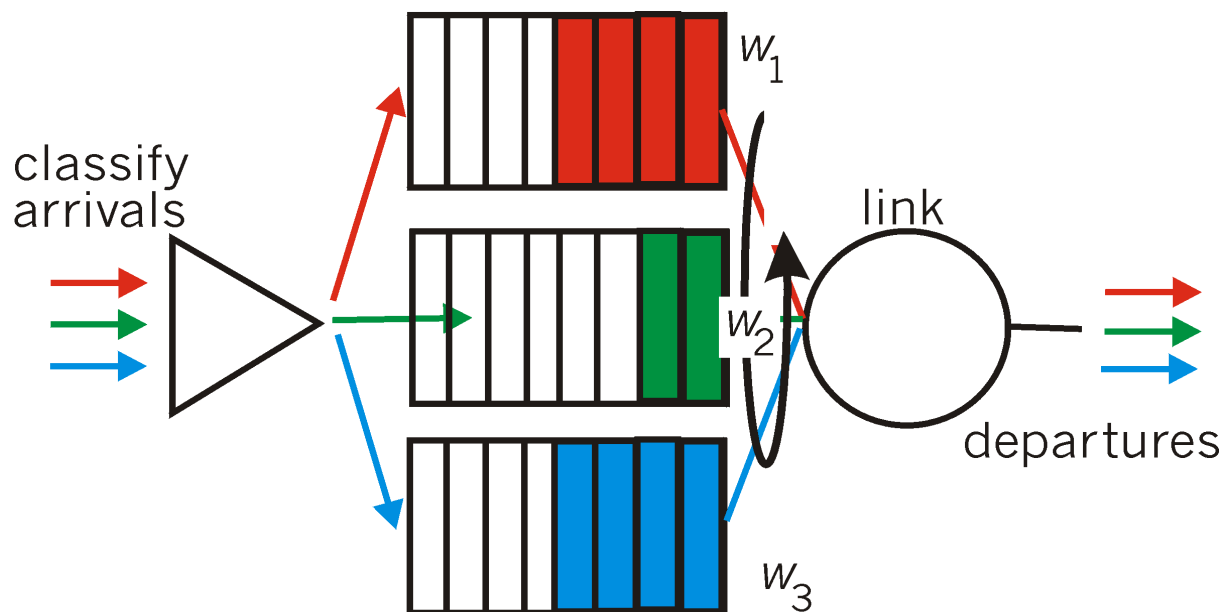
- 按类别排队
- 循环扫描各类别的队列, 每个队列服务一次 (若队列不空)



调度机制: (续)

加权公平调度 Weighted Fair Queuing或
generalized Round Robin

- 为每个类别的队列分配一个服务权重 w_i





加权公平调度 WFQ

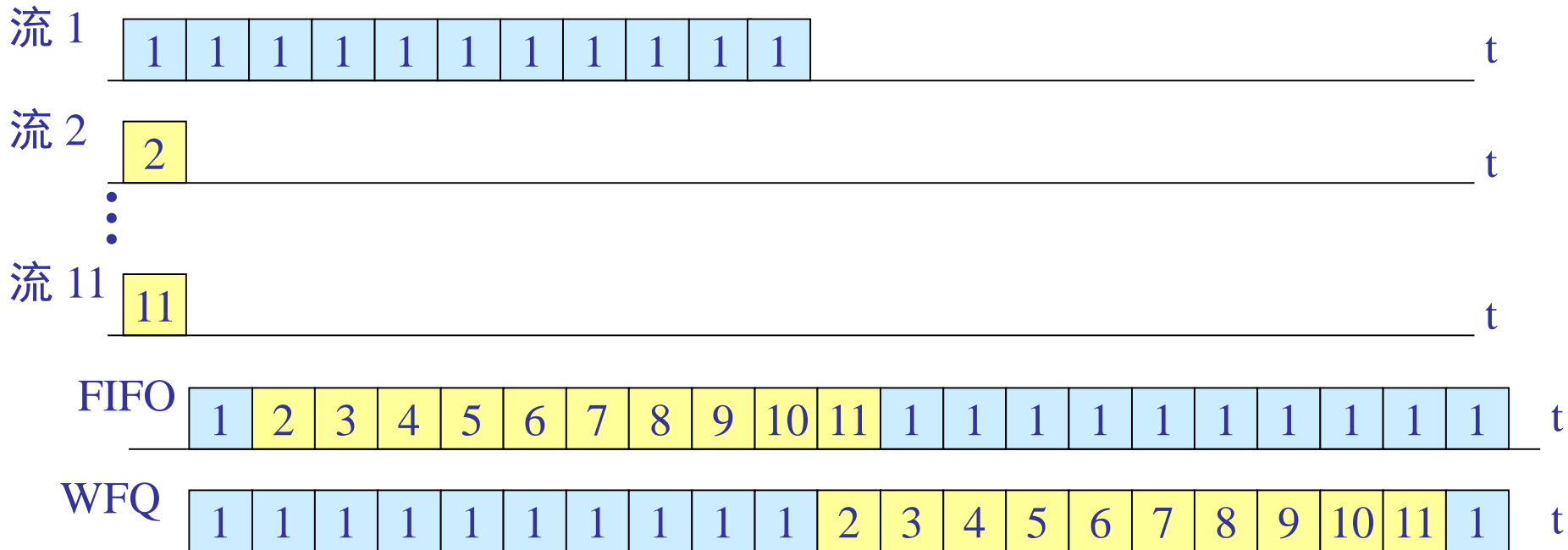
- 对分组分类，存储到相应的队列中
- 给队列*i*指派一个权重 w_i 。队列*i*得到的平均服务时间为 $w_i / (\sum w_j)$ ， $\sum w_j$ 是对所有非空队列的权重求和
- 队列*i*得到的带宽 R_i 为
$$R_i = \frac{R \times w_i}{\sum w_j}$$
- 算法：定义 a_i ， L_i ， f_i 分别为第*i*个分组的到达时刻、分组长度及发送结束时刻，则计算每个队列中的 $f_i = \max(f_{i-1}, a_i) + L_i / R_i$ ，选择发送最小 f_i 的队列的分组

WFQ 与 FIFO 的比较

(a) 分组流 1 的分组连续输入

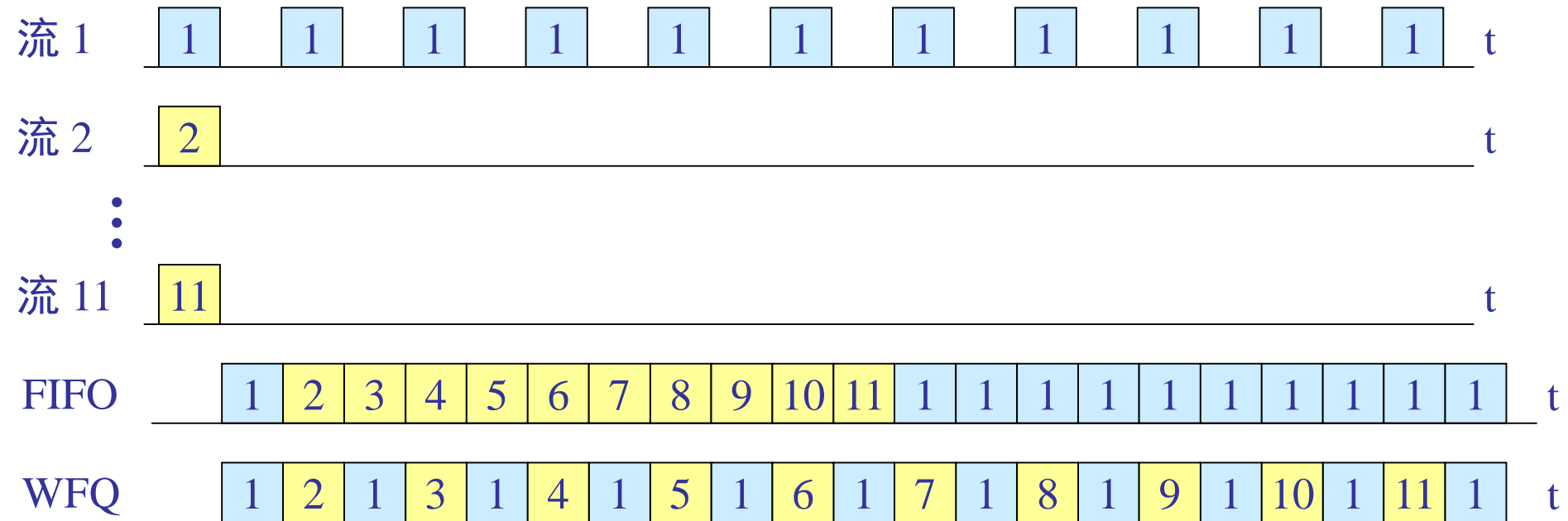
$$W_1=0.5$$

$$W_i=0.05, i=2,3,\dots,11$$

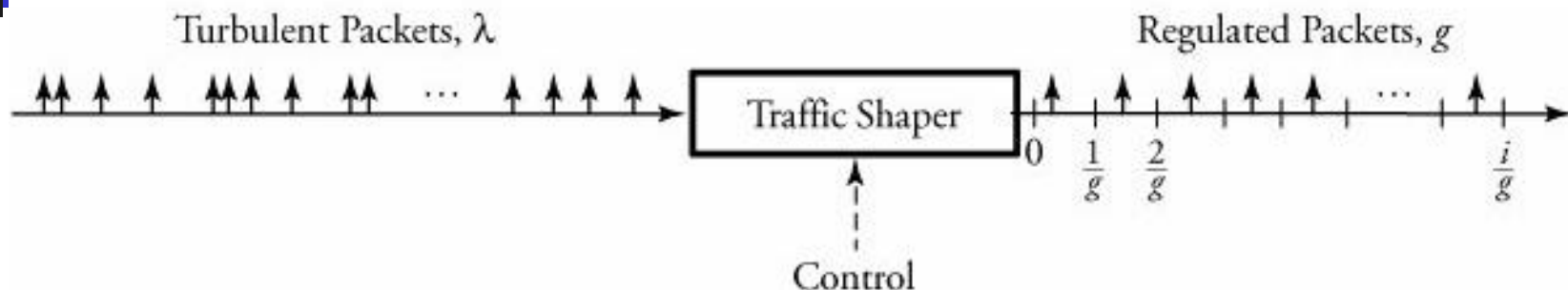


WFQ 与 FIFO 的比较

(b) 分组流 1 的分组断续输入



流量整形 (Traffic Shaping)



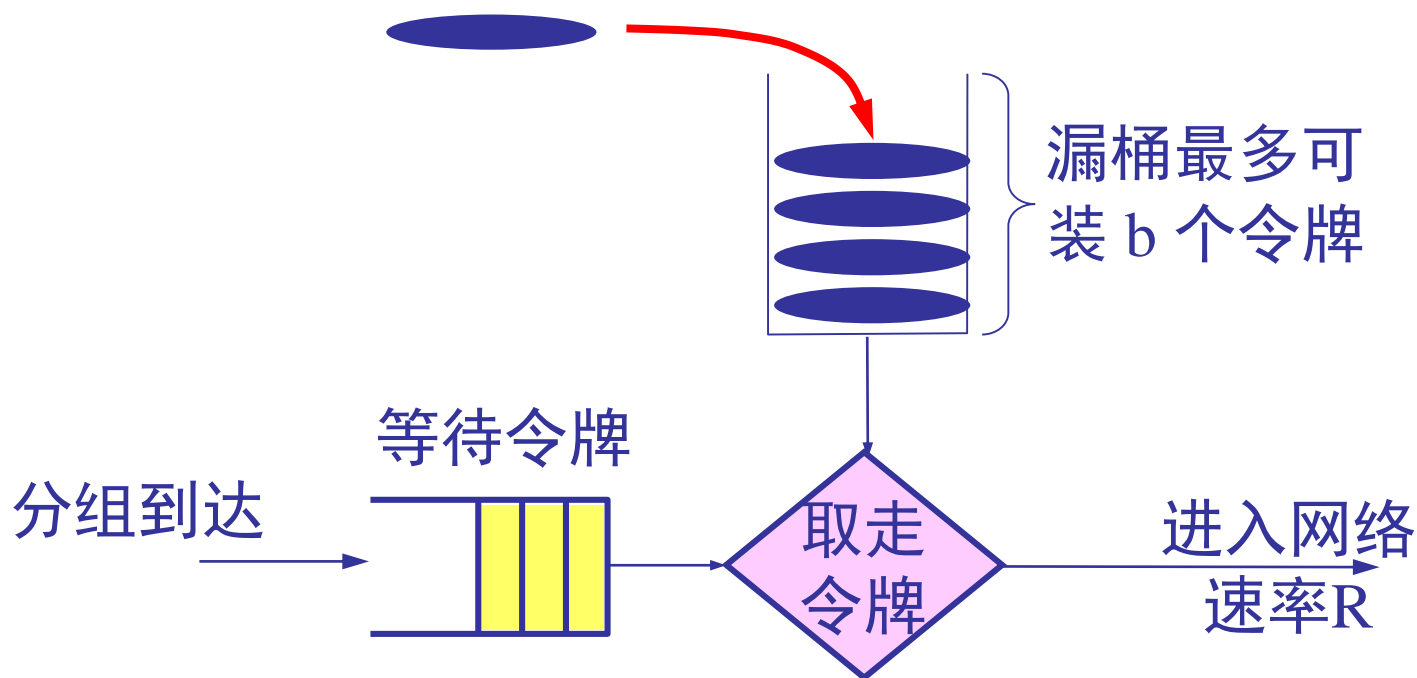
目的：将输入流的速率控制在有效带宽之内，以避免拥塞及分组延迟

- (1) 平均速率：控制一个数据流的平均速率
- (2) 峰值速率：限制数据流在非常短时间内的流量
- (3) 突发长度：限制在非常短时间内连续注入到网络中的分组数

流量整形：令牌桶(Token bucket)

流量整形：调节进入网络的数据流发送速率，使之不超过突发长度及平均速率

注入漏桶的令牌(token) 为每秒 r 个



在时间 t 内允许进入网络的分组数 $\leq r t + b$



流量整形

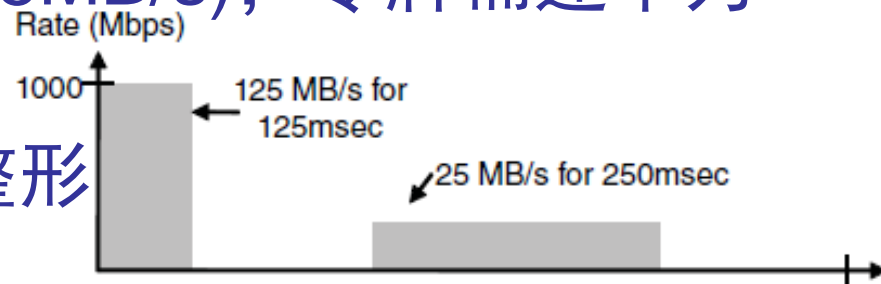
- 突发时间: $rt + b = tR, \quad t = b/(R-r)$
- 突发数据长度: $Rt = Rb/(R-r)$
- 一个流的长期速率由 r 限制, 短期突发长度由 b 决定
- 令牌桶算法的应用
 - 对注入网络的分组整形及监管, 即调节速率及突发长度
 - 主机利用操作系统提供的令牌桶算法控制发送速率
 - 路由器在网络接口上用令牌桶算法监管流量

流量整形

主机速率为1000Mbps(=125MB/s)，令牌桶速率为200Mbps (=25MB/s)

主机输出的流经过令牌桶整形

(a) 主机输出的流



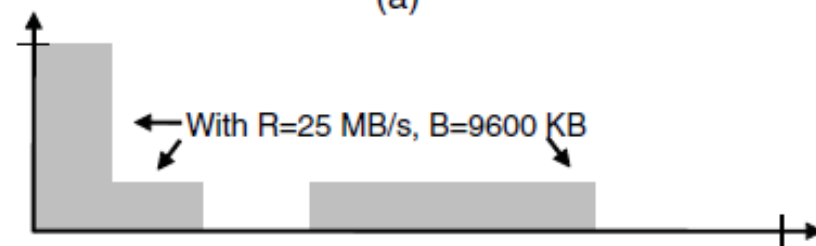
(a)

(b) $b=9600\text{KB}$ ，有短时突发

突发时间： $t=b/(R-r)$

$$=9.6/(125-25)$$

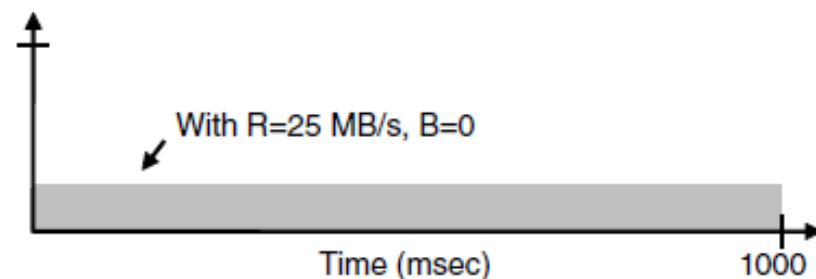
$$=0.096\text{ s}$$



(b)

突发长度： $Rt=125*0.096$

(c) $b=0\text{ KB}$ ，无突发



(c)

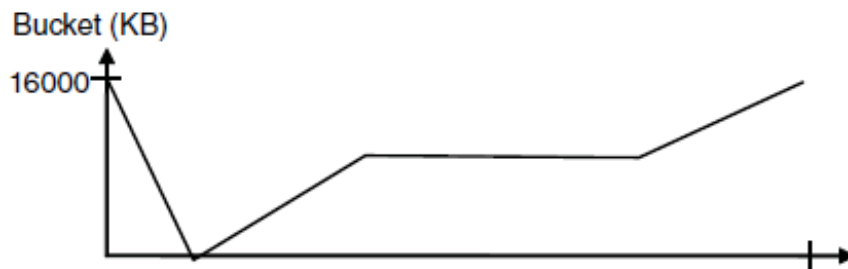
流量整形

通过200Mbps (=25MB/s)
的令牌桶整形输出，令牌
桶的高度不同，突发数据
量也不同

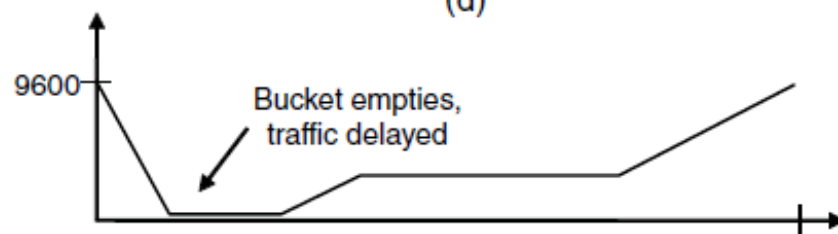
(d) $b=16000$ KB

(e) $b=9600$ KB

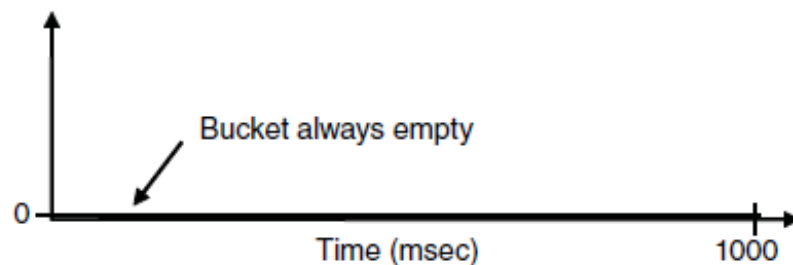
(f) $b=0$ KB



(d)



(e)



(f)

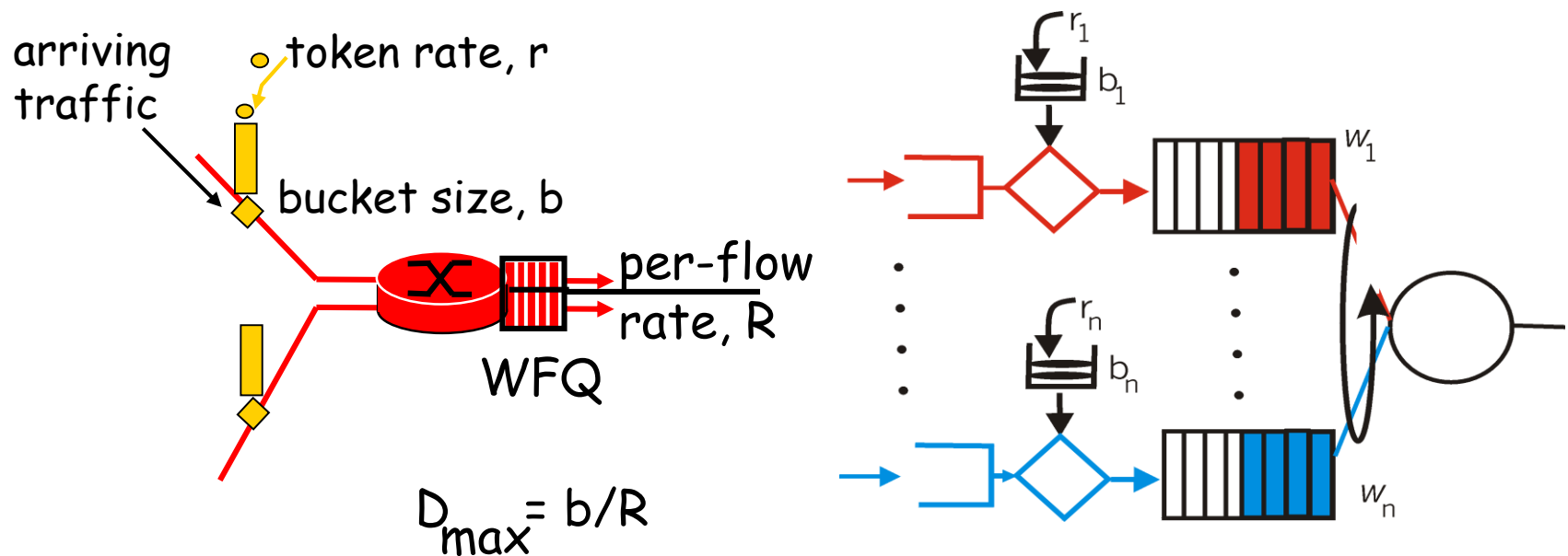


漏桶与令牌桶的区别与联系

- 令牌桶： $B > 0$ ，在限制平均速率的同时还允许某种程度的突发
- 漏桶： $B = 0$ ，强行限制数据速率，不允许突发
- 漏桶算法不能有效地使用网络资源。因为漏桶的发送速率是固定的，即使网络中没有发生拥塞，漏桶算法也不能使某一个单独流达到端口速率。因此，漏桶算法对于存在突发特性的流量来说缺乏效率。而令牌桶算法则能够满足这些具有突发特性的流量。
- 漏桶算法与令牌桶算法结合为网络流量提供高效的控制。

令牌桶机制与加权公平调度结合

- 第1级令牌桶： r 、 b 的值较大，平滑流量
 - 分组经第1级缓存，最大时延为 b/R
- 第2级漏桶： $b_1=b_n=0$ ，禁止突发，用 r_1, r_2 调节每个流的速率

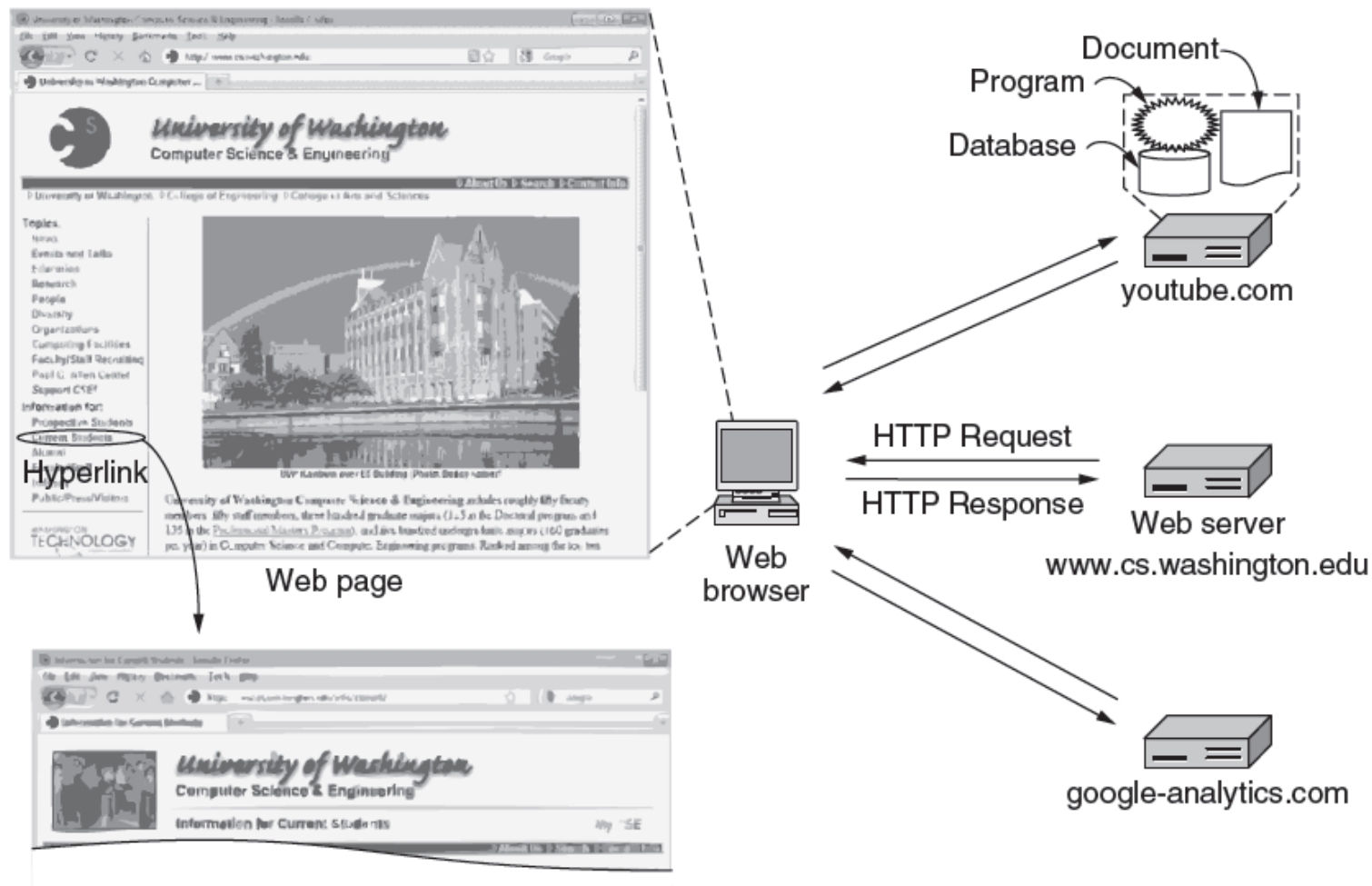




QoS概述

- 理解QoS
- QoS机制：
 - RSVP
 - DiffSer模型
 - 调度机制与漏桶算法
- 其他网络技术
 - Web与HTTP
 - CDN：内容分发网络
 - MPLS：多协议标记交换

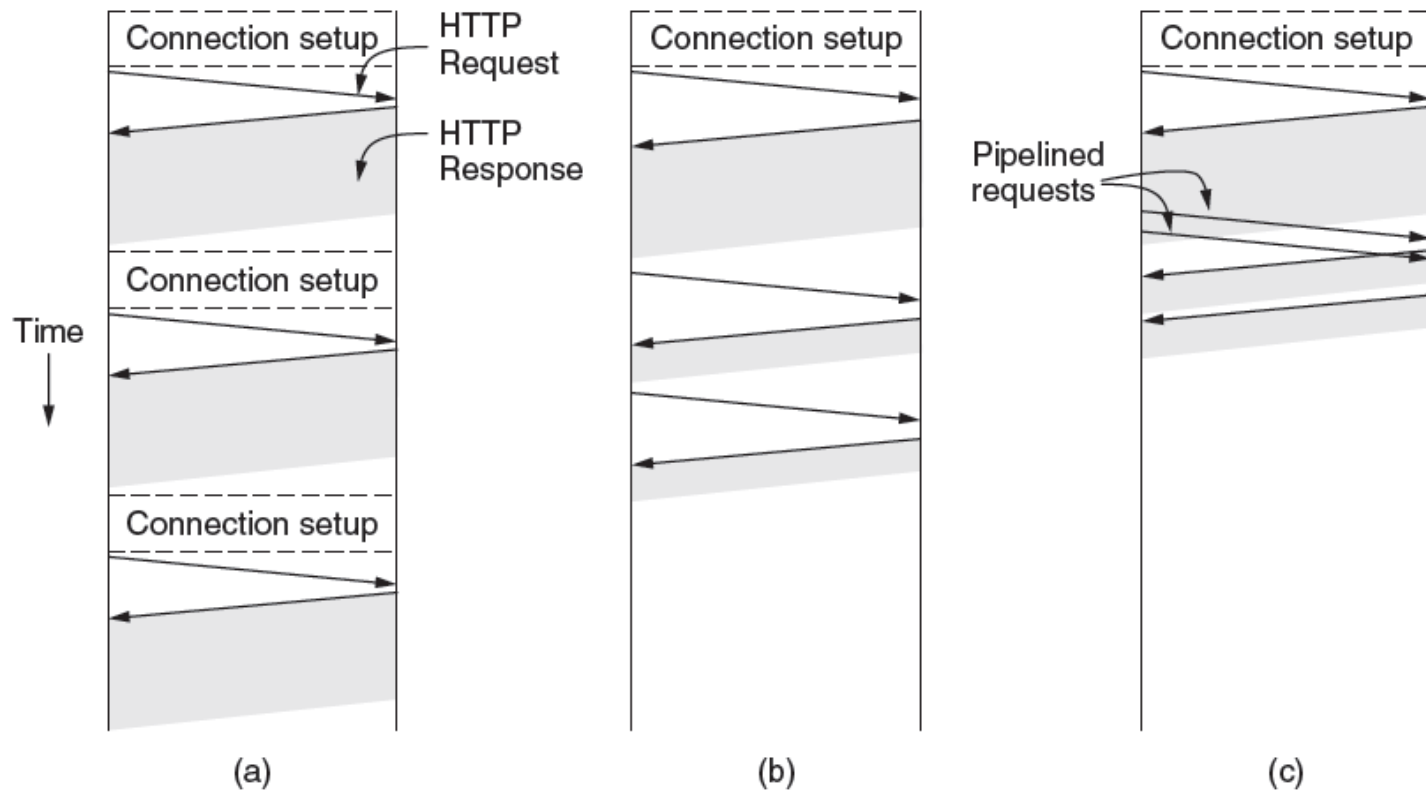
Web体系结构概述(1)



- Web页面（HTML，超文本、超链接），URL（统一资源定位符）
- 浏览器与服务端之间的信息传输：HTTP

HyperText Transfer Protocol (1)

- (a) 具有多个连接和系列请求. HTTP1.0
- (b) 具有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, 无需慢启动
- (c) 具有一个持续连接和流水线式请求 HTTP1.1 连接重用





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



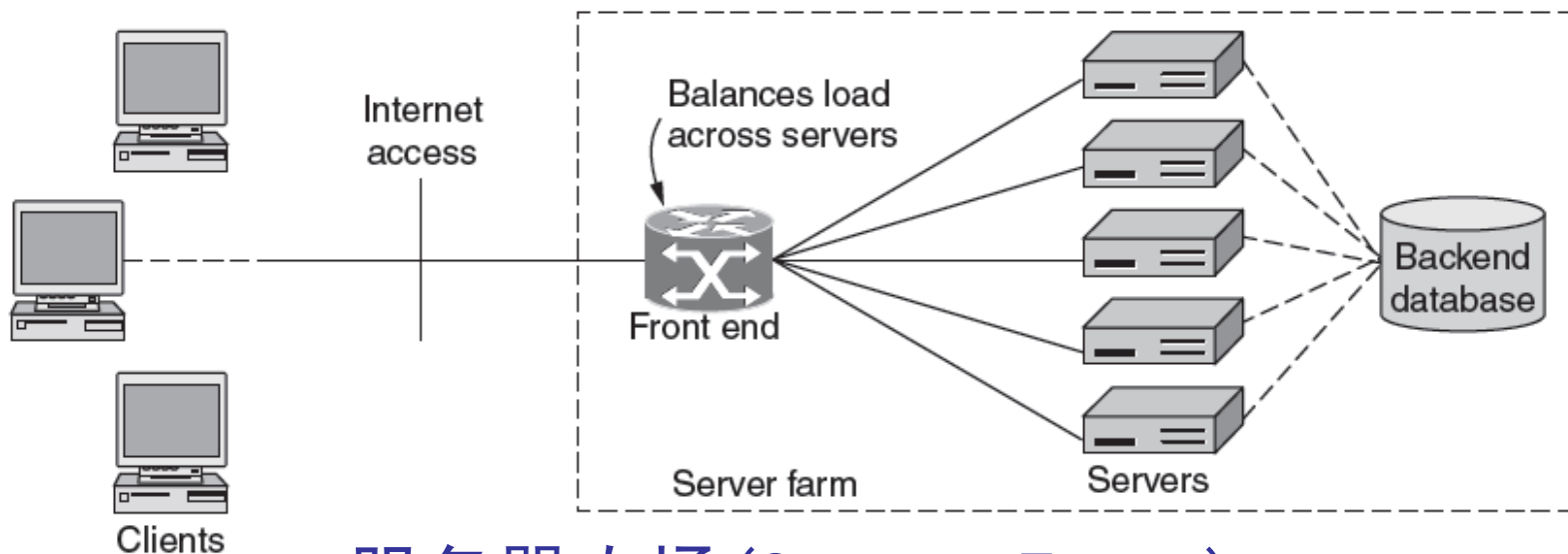
HyperText Transfer Protocol (3)

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

内容分发：服务器农场和Web代理 (1)

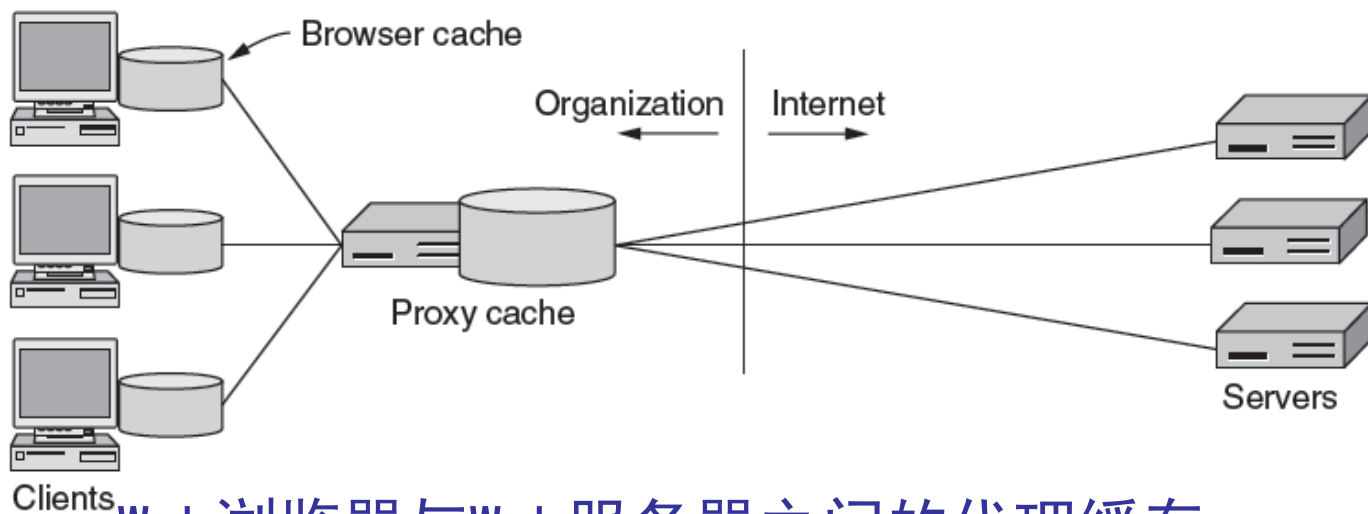
- 构建大型Web站点并提供良好性能，需要加快服务端或客户端的处理速度。
- 客户端采用更好的缓存技术：页面缓存，降低流量
- 服务器端采用服务器集群或服务器农场
 - 多台计算机作为Web服务器与共同的后端数据库连接，使得经过任意一个服务器所获取的数据是一致的



服务器农场 (Server Farms)

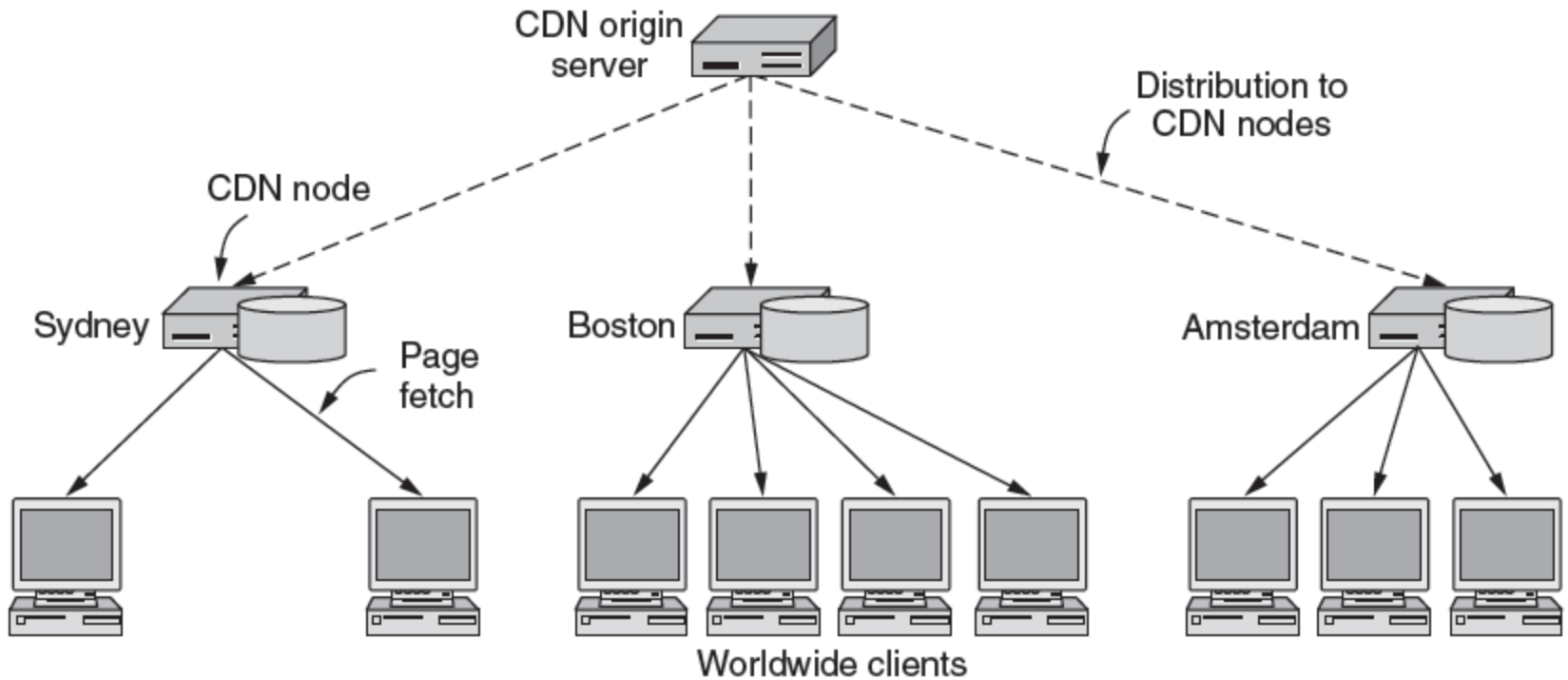
内容分发：服务器农场和Web代理(2)

- 前端将入境请求分发到服务器缓存池中：由路由器或交换机根据窥探的IP、TCP及HTTP头部数据，确保将同一TCP连接报文转发给同一个服务器，如利用Web Cookie
- 减少服务器访问频度：除浏览器缓存之外，Web代理使多用户共享缓存（页面的二级缓存），如为某一组织设置一个代理，减少流量以及响应时间
- 使用DNS重定向，实现负载均衡



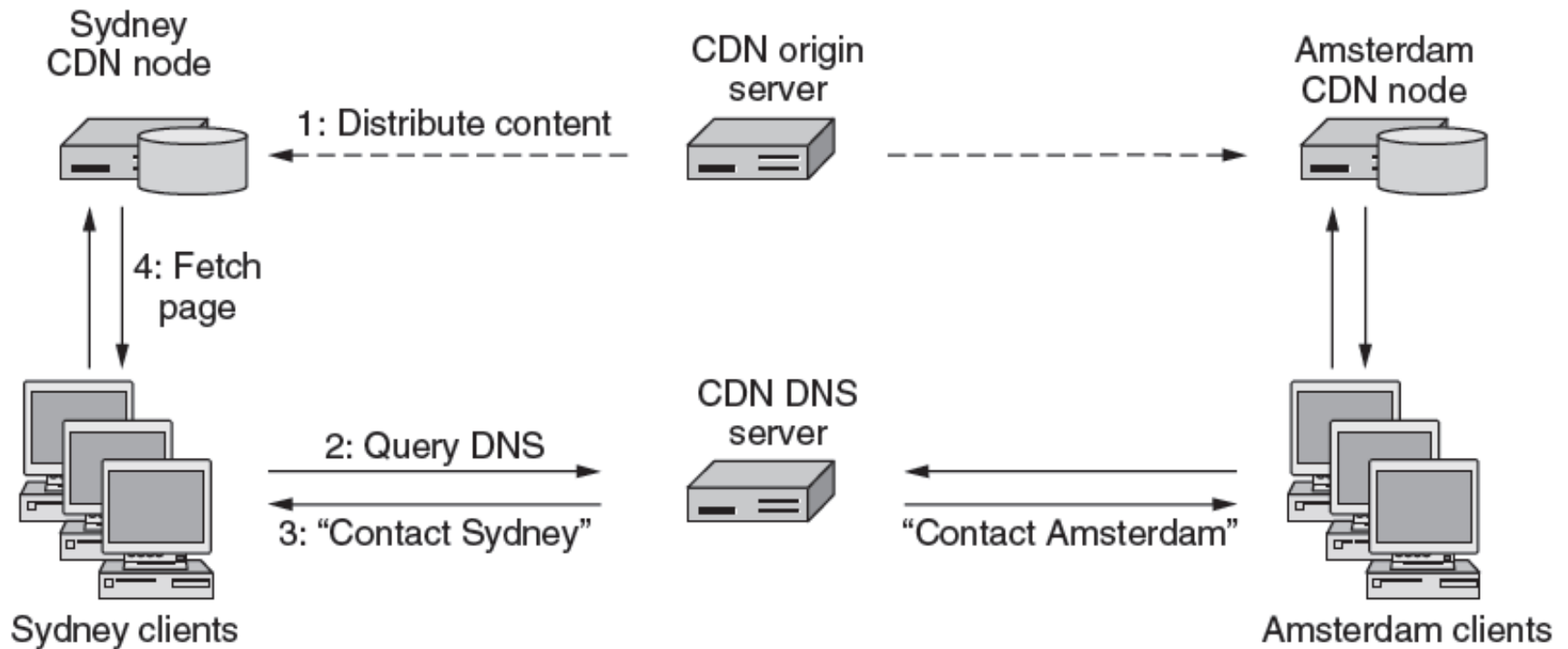
Web浏览器与Web服务器之间的代理缓存

Content Delivery Networks (1)



CDN 分发树

Content Delivery Networks (2)



CDN 节点利用DNS将客户指向附近的节点



Content Delivery Networks (3)

```
<html>
<head> <title> Fluffy Video </title> </head>
<body>
<h1> Fluffy Video's Product List </h1>
<p> Click below for free samples. </p>

<a href="koalas.mpg"> Koalas Today </a> <br>
<a href="kangaroos.mpg"> Funny Kangaroos </a> <br>
<a href="wombats.mpg"> Nice Wombats </a> <br>
</body>
</html>
```

(a) 原始的Web页面

```
<html>
<head> <title> Fluffy Video </title> </head>
<body>
<h1> Fluffy Video's Product List </h1>
<p> Click below for free samples. </p>

<a href="http://www.cdn.com/fluffyvideo/koalas.mpg"> Koalas Today </a> <br>
<a href="http://www.cdn.com/fluffyvideo/kangaroos.mpg"> Funny Kangaroos </a> <br>
<a href="http://www.cdn.com/fluffyvideo/wombats.mpg"> Nice Wombats </a> <br>
</body>
</html>
```

(b) 与CDN连接的页面

(b)



CDN

路由选择：如何选择最好CDN服务器？

- CDN 建立一张“映射表”，指示ISP到CDN节点的距离
- 当请求到达授权DNS服务器时：
 - 服务器决定来自用户的查询来自哪个ISP
 - 使用“映射表”，决定最好的CDN服务器
- CDN建立的应用覆盖了整个网络



QoS概述

- 理解QoS
- QoS机制：
 - RSVP
 - DiffSer模型
 - 调度机制与漏桶算法
- 其他网络技术
 - Web 与 HTTP
 - CDN: 内容分发网络
 - MPLS: 多协议标记交换

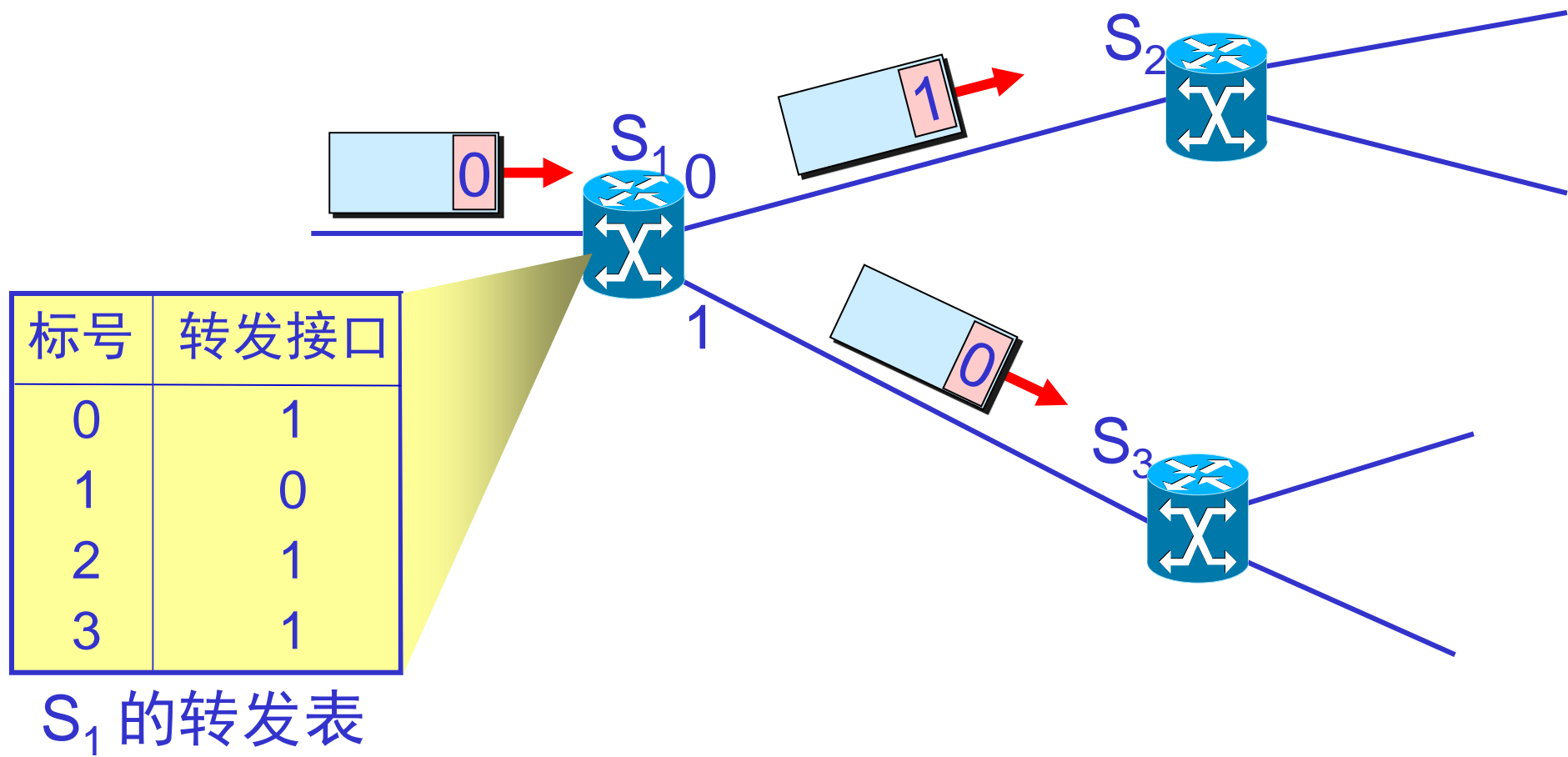


多协议标记交换MPLS

- MPLS(MultiProtocol Label Switching): 用面向连接的方式代替IP的无连接分组交换, 利用更快捷的查找算法, 而不用最长前缀匹配的方法来查找路由表
- 在传统的路由器上也可以实现MPLS
- MPLS 的特点
 - (1) 支持面向连接的服务质量
 - (2) 平衡网络负载
 - (3) 支持虚拟专用网VPN

MPLS 的工作原理

每个分组携带一个标记(label)——一个整数，交换机读取分组的标记，并用标记值来检索转发表

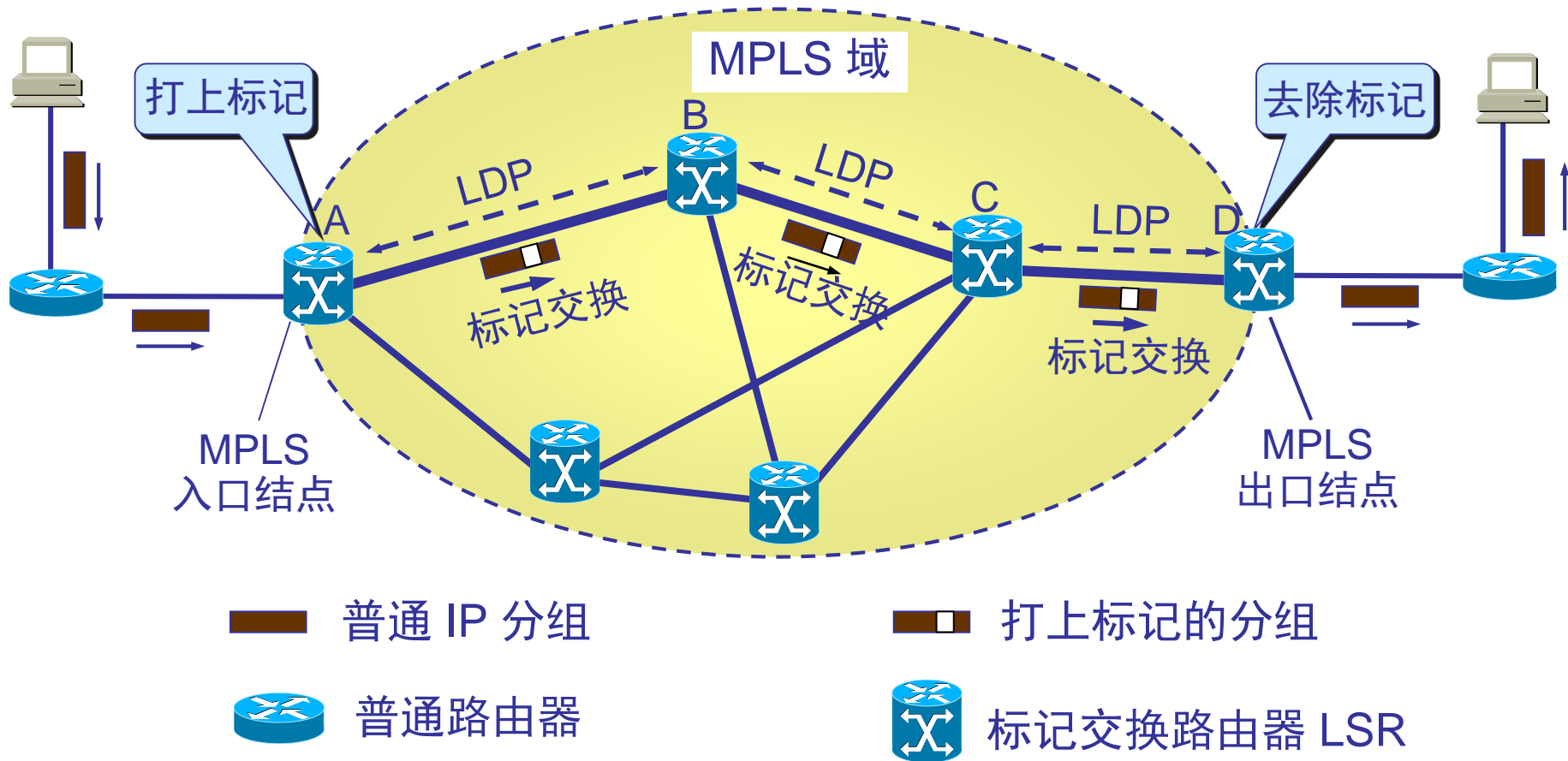




MPLS 的工作原理

- MPLS 对打上固定长度“**标记**”的分组用**硬件**进行转发，节省查找路由表的时间，分组转发的速率加快
- 采用硬件技术对打上标记的分组进行转发称为**标记交换**。“**交换**”表示**根据第二层的标记用硬件进行转发**，不再是在第三层上分析IP首部和查找转发表
 - 类似于VLAN IEEE802.1Q采用的方法

MPLS 协议的基本原理





MPLS 的工作过程

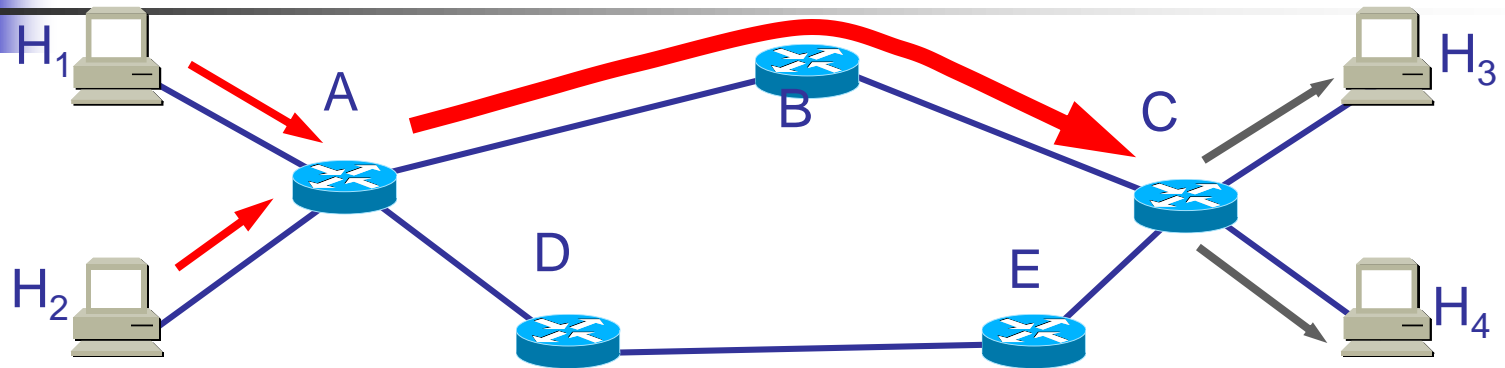
- (1) MPLS 域中的标记交换路由器（LSR）之间采用标记分配协议LDP交换报文，找出标记交换路径LSP，各LSR根据路径构造分组转发表
- (2) 分组进入MPLS域时，由MPLS入口结点**打上标记**，并按照转发表转发给下一个LSR
- (3) 所有LSR都按照标记转发：每经过一个LSR，
换一个新的标记；
 - 类似于虚电路方式
- (4) 分组离开MPLS域时，由MPLS 出口结点**去除标记**；之后，按照一般分组的转发方法转发



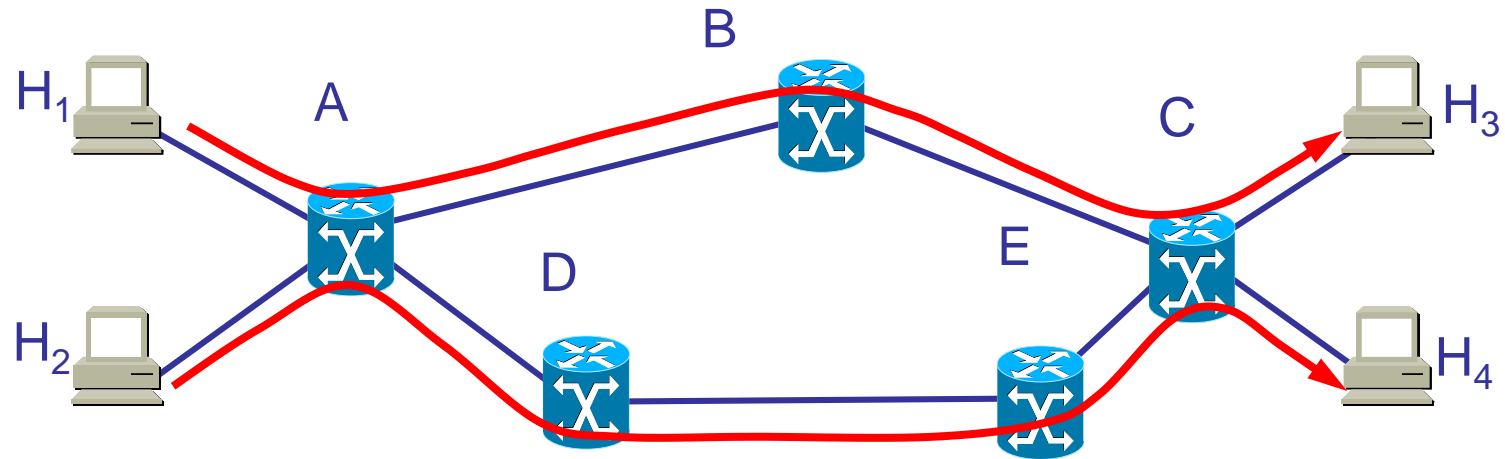
转发等价类 FEC

- FEC: (Forwarding Equivalence Class)
“转发等价类”，路由器按照同样方式对待的分组的集合
- 划分 FEC 的方法由网络管理员来控制，因此非常灵活
- 入口结点将属于同样 FEC 的分组指派同样的标记，FEC与标记一一对应

FEC 用于负载平衡

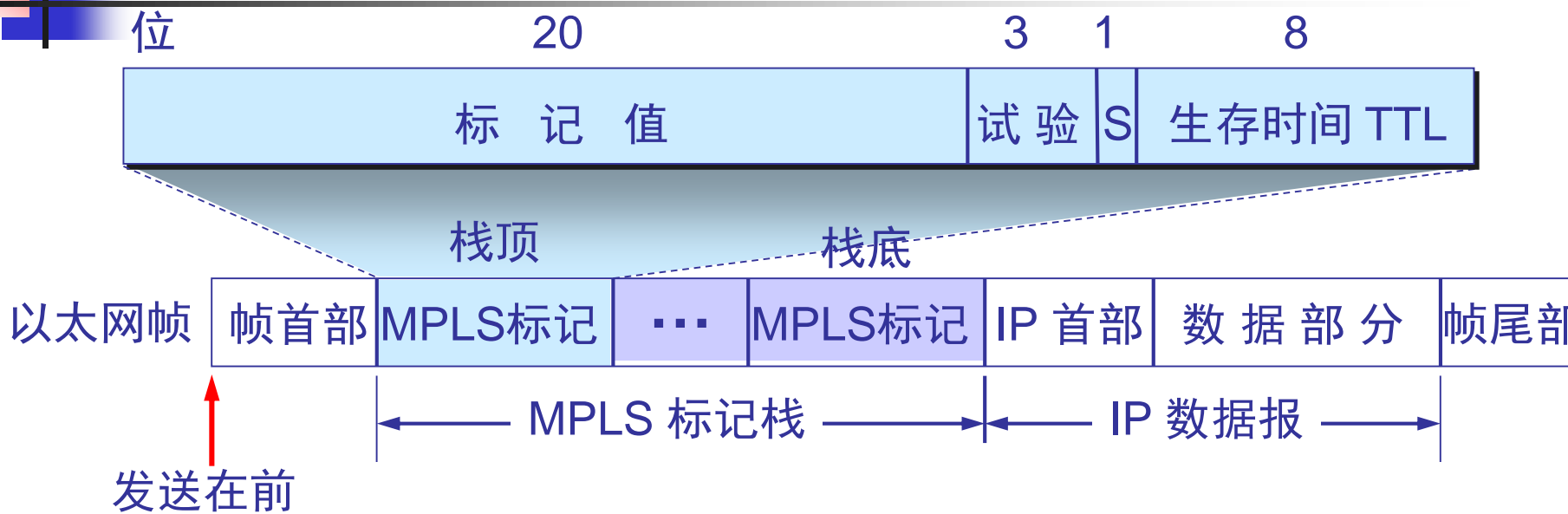


(a) 传统路由选择协议，基于目的地址选择最短路径，可能导致A→B→C 过载



(b) 基于FEC设置标记，利用FEC使通信量分散

MPLS 首部



- MPLS 标记可以有多个，采用栈结构；
- 一旦产生就压入标记栈；
- 最新的标记在帧首部之后，便于硬件在固定位置读取标记



小结：多媒体业务与服务质量

- 多媒体网络应用
- 支持多媒体业务的协议
 - RTP, RTCP, RTSP, SIP
- 保证QoS技术
 - 综合服务：RSVP与接纳控制
 - 区分服务
 - 流量整形与漏桶机制
- 其他网络技术
 - Web 与 HTTP
 - CDN：内容分发网络
 - MPLS：多协议标记交换



练习题

- 在6Mbps的网络上，一台主机的输出流通过令牌桶整型。令牌桶的速率为1Mbps，初始令牌桶被填满到8MB。试问该计算机以6Mbps速率发送，持续时间是多少？
- 你如何理解QoS？MPLS的工作原理是什么？MPLS对于保证QoS有何作用？

练习题

- 设B向D用UDP发送一个长文件。为防止网络拥塞，路由器R1对流量监管。R1先使用令牌桶给用户以较高速率突发少量数据，随后再用漏桶进一步平滑流量。假设令牌生成速率为20MBps，令牌桶的容量为30MB，漏桶排空速率为30MBps，路由器最高速率为50MBps。试问（1）当用户突发数据量为200MB且令牌桶为满时，令牌桶的输出流量形式和漏桶的输出流量形式（以多高速率输出多少时间表示）。（2）用户最终获得的突发速率为多少？

