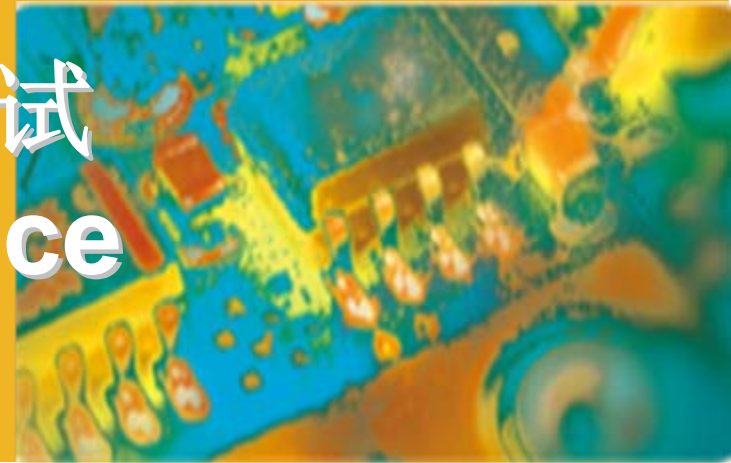




温州大學  
WENZHOU UNIVERSITY

# 第7章 网络服务质量测试

## QoS: Quality of Service

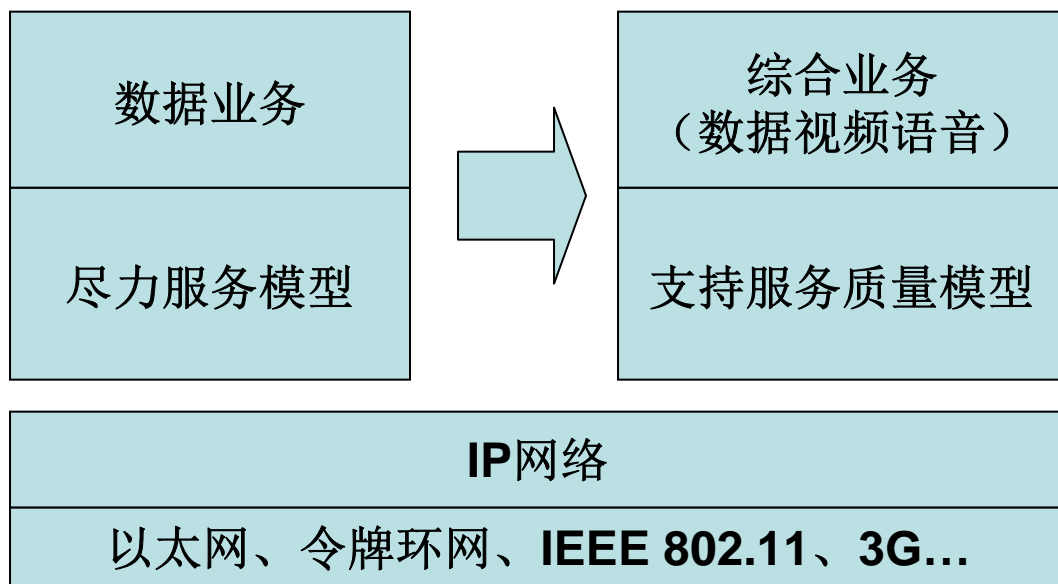


<http://network.wzu.edu.cn>  
<http://www.spirent.com>  
[linflag@163.com](mailto:linflag@163.com)

# 内容提要

- ✓ QoS测试的必要性
- ✓ QoS概述
- ✓ QoS模型
- ✓ QoS中的队列调度与管理技术
- ✓ 网络服务质量测试方法学

# 背景



**IP**将成为未来各种网络技术和业务的融合平台

# QoS的应用需求

- ✓ 网上实时传输多媒体数据，以便不同用户在不同设备上能够共享图像、声音、影视等多媒体资源，以及进行“面对面”的交流。其中
  - (1) “**多媒体数据(multimedia data)**”是指组合文字、图像、声音和视像的数据，尤其是声音和影视数据
  - (2) “**实时传输**”可简单理解为接收数据几乎与发送数据同时完成，如现场实况广播就是属于这种情况
- ✓ 在网络上的多媒体应用和数据通信应用的差别
  - 前者要求在客户端播放声音和图像时要流畅，声音和图像要同步，因此对网络的时延和带宽要求很高
  - 后者则要把可靠性放在第一位，对网络的时延和带宽的要求不那么苛刻

# QoS的应用需求

各种应用所要求的服务质量

应用	可靠性	时延	抖动	吞吐率/带宽
IP电视会议	低	小	小	高
IP电话	低	小	小	低
IP电视	低	-	小	高
影视点播(VOD)	低	-	小	高
音乐点播(AOD)	低	-	小	中
Web访问	高	中	-	中
文件传输	高	-	-	中
电子邮件	高	-	-	低

# QoS的应用需求

## 多媒体服务质量参考值

### 视听应用的服务质量参数(参考值\*)

媒体	应用	互动方式	数据速率 举例	关键参数和目标值			
				单向时延	抖动	丢包率	吞吐率不低于
声音	IP电话	双向	4~32 kbps	150~400	<1ms	<3%	4~32 kbps
视像	影视点播	单向为主	30 Mbps	<10s	<5ms	<3%	30 Mbps

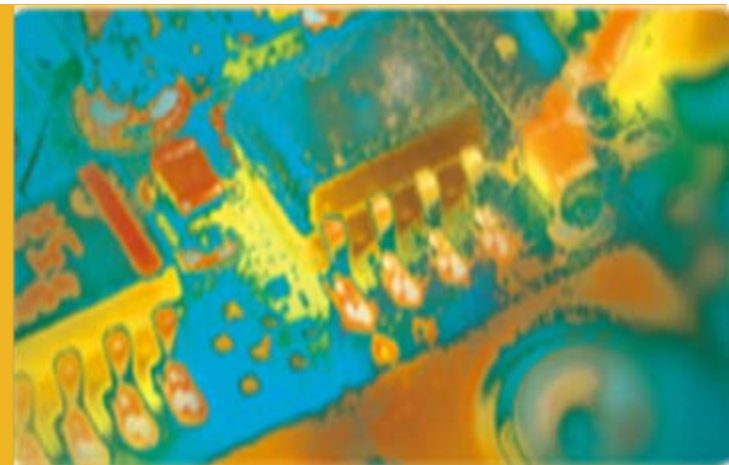
# QoS测试的必要性

- ✓ 需要在IP网络上提供不同级别的QoS，以提供具有服务质量的传输。
- ✓ QoS测评的必要性：
  - (1) 用户
    - 检查用户端QoS能否满足用户需求
  - (2) 网络运营商
    - 对网络进行QoS性能测试
  - (3) 设备研发机构
    - 设备能否根据设定的服务质量要求进行数据转发



温州大学  
WENZHOU UNIVERSITY

# 第7章 网络服务质量 测试



## 7.2 QoS概述



# QoS的基本概念

- ✓ 根据RFC2386中的描述，服务质量QoS(Quality of Service)定义为网络在传输数据流时要求满足的一系列服务请求，具体可以量化为带宽、时延、时延抖动、吞吐量和丢包率等性能指标。
- ✓ 强调传输服务在端到端（end-to-end）或者网络边界到网络边界范围的整体性。
- ✓ 服务质量的最终目的就是为各种业务提供可靠的端到端的服务质量保证。
- ✓ 在网络中通过业务等级协定（Service Level Agreement, 简称SLA）与客户进行相互协调以确定相应的服务质量等级。

# QoS的基本概念

## ✓ QoS的两种解释

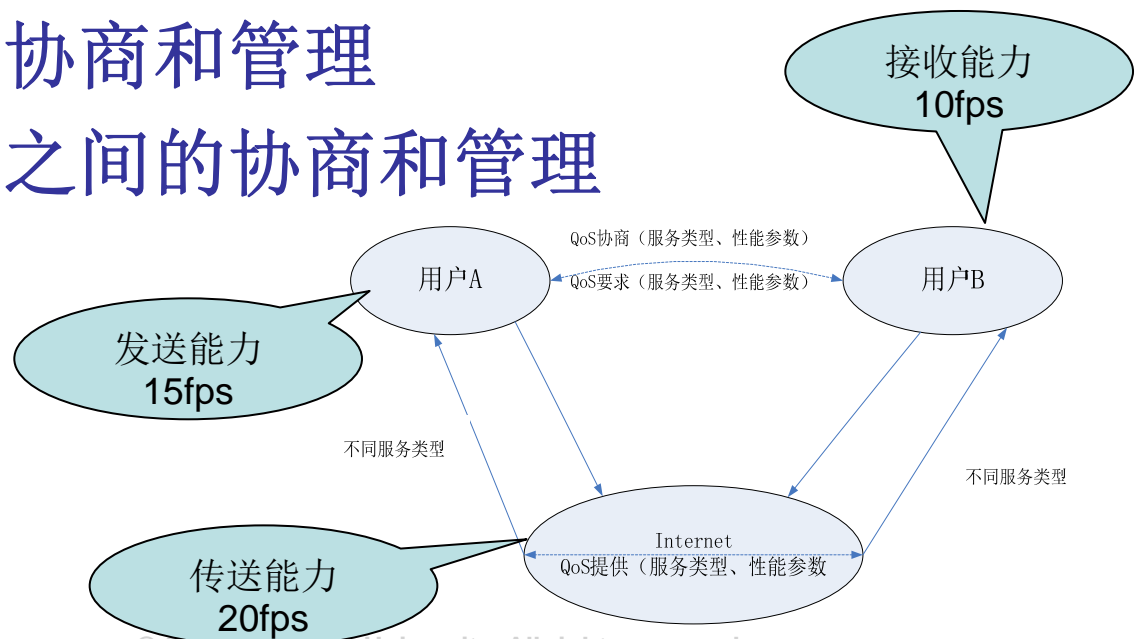
- 从服务角度来看，参照ITU推荐标准X.902，QoS是指媒体应用(如IP电话)对网络的交通管理和传输性能提出的需求。交通管理主要体现在管理软件上，传输性能主要体现在硬件上。不同的应用对服务质量有不同的要求，需要不同的交通管理方法。
- 从技术角度来看，QoS是保障网络按不同要求运行的控制方法(mechanism)，也称策略、机制或技巧。能够保障要求的服务称为保障服务(guaranteed service)，不能保障但可预测的服务称为预测服务(predictable service)，不能完全保障但可达到部分要求的服务称为“尽力服务(best effort service)”

## ✓ 服务质量的高低取决于

- 执行数据传输控制策略的软硬件
- 网络本身的性能

# QoS的基本概念

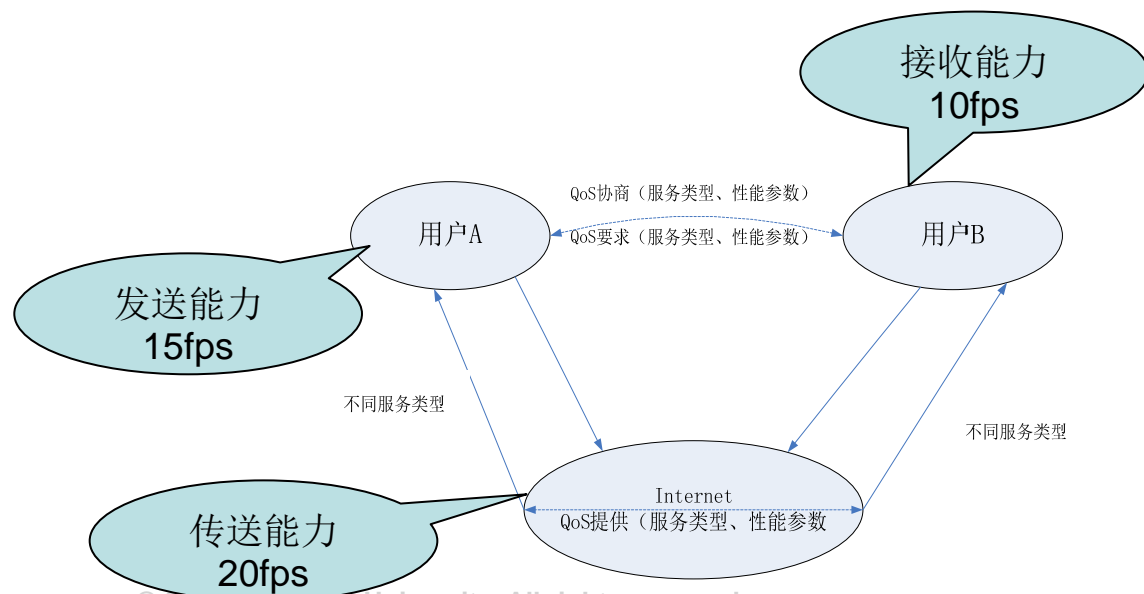
- ✓ 提供端到端的性能保证绝非易事，它需要网络体系结构中所有各层以及端到端之间所有的**网络元素**的整体协作。
  - 用户之间的协商
  - 用户和网络的协商和管理
  - 网络各个节点之间的协商和管理



# QoS的基本概念

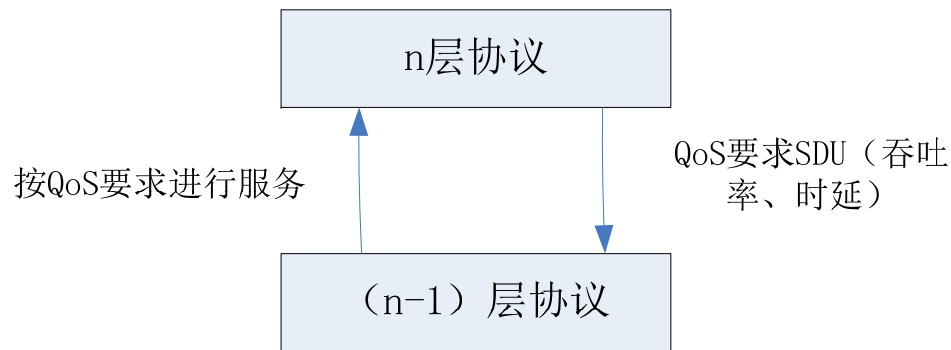
✓ 节点应该具有以下功能：

- 1、各个节点应能够识别各个不同级别的业务。
- 2、为满足高优先级业务的资源需求，各个节点或者能够进行网络资源的预约，或者具有相应的资源调度算法。
- 3、为尽量保证高优先级业务不发生丢弃，满足其时延、抖动要求，各个节点应有合理的排队算法和队列管理调度算法。



# 端到端QoS保证

- ✓ 端到端通信的概念 :源主机进程到目的主机进程之间的通信
- ✓ IP网络端到端的QoS就必须从网络体系结构出发, 要求进行逐层保证

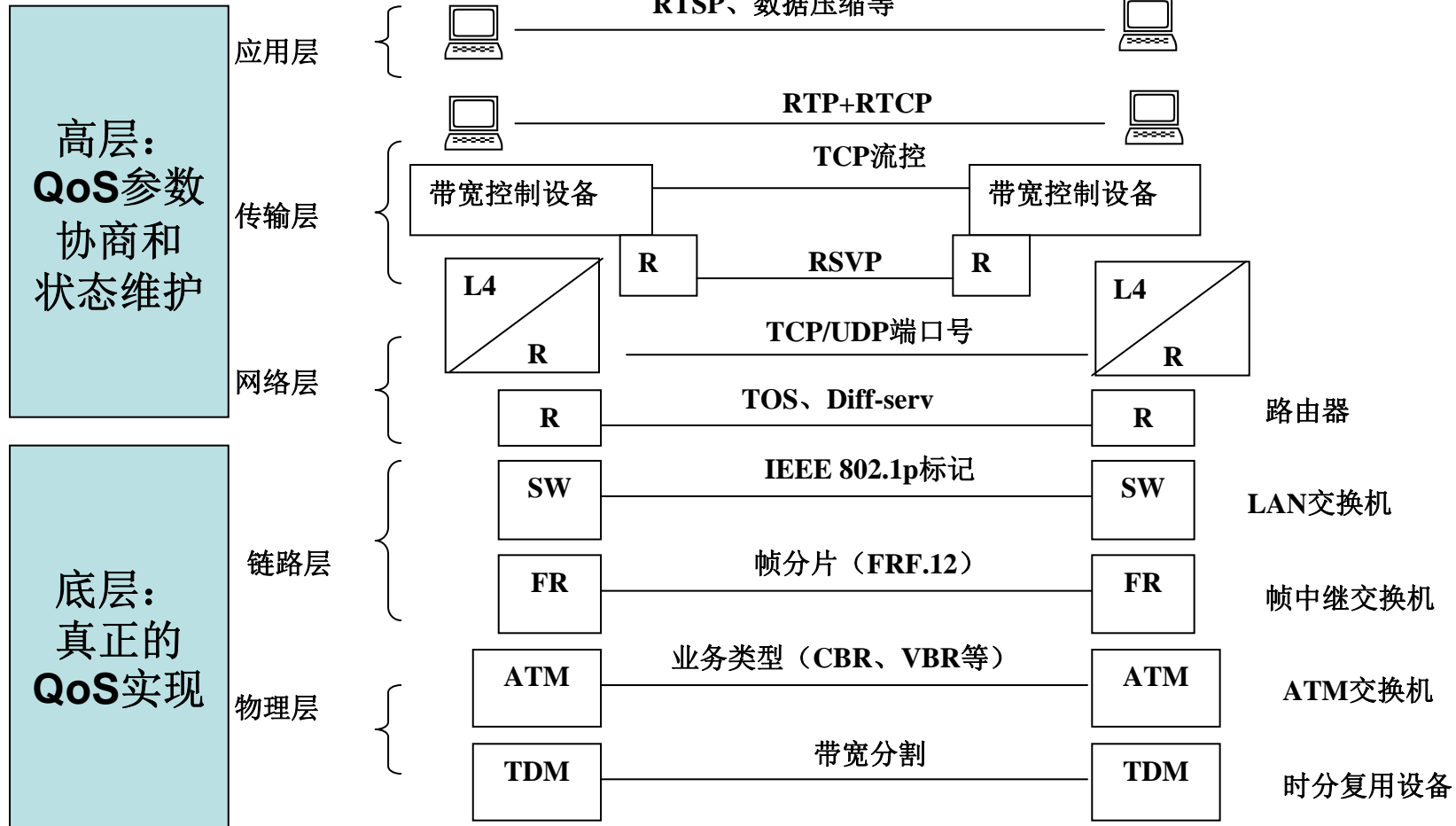


# 端到端QoS保证

## ✓ TCP/IP中与QoS有关的内容:

- 1) 数据链路层: **802.1p**规范使得第二层交换机能够提供流量优先级和动态组播过滤服务, **802.1P** 协议头包括一个**3位优先级字段**, 该字段支持将数据包分组为各种流量种类。**802.1P** 流量被简单分类并发送至目的地, 而没有带宽预留机制。
- 2) 网络层: **IP**数据包首部的**ToS (Type of Service)** 字段, 用于标记数据包的优先级。
- 3) 传输层: **端口号**作为传输层提供给应用层的服务访问点, 从**QoS**角度而言, 正是用于区分不同的应用服务。

# 按层次对QoS技术分类



**CBR: Constant Bit Rate**

**FRF: Frame Relay Forum**

**RTP: Real-time Transport Protocol**

**VBR: Variable Bit Rate**

**TOS: Type Of Service**

**RTCP: RTP Control Protocol**

**UBR: Unspecified Bit Rate**

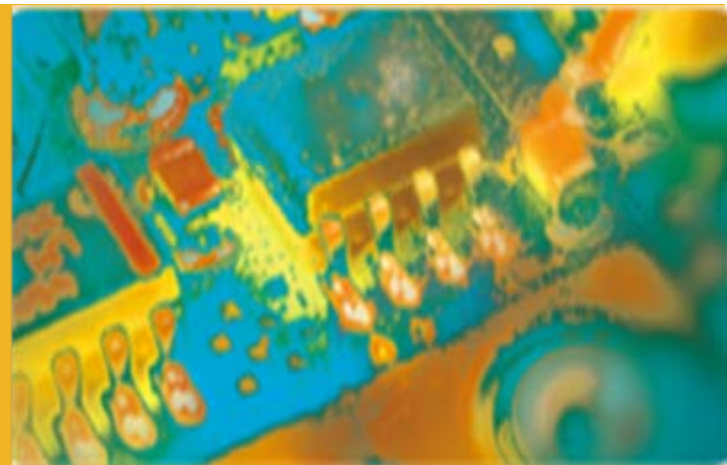
**Diff-serv: Differentiated service**

**RTSP: Real Time Streaming Protocol**



温州大學  
WENZHOU UNIVERSITY

# 第7章 网络服务质量 测试



## 7.3 QoS模型



# QoS模型

- ✓ **尽力而为服务(Best-Effort Service)**：“尽力而为”的工作方式，不保证服务质量----**无QoS**
- ✓ **综合服务(IntServ)模型**：采用信令机制，通知数据流所经由的路径上预留资源，可以实现很好的质量保证。但当网络规模很大时开销也很大，一般适于在网络边缘使用----**信令化/动态的QoS**
- ✓ **区分服务(DiffServ)模型**：把有相似要求的业务分为类，按类提供不同的服务质量的方式。不需要专门信令的支持，可根据处于边缘网络还是核心网络，采取不同的工作方式相协调----**预先部署的QoS**
- ✓ **Diff-IntServ综合模型**

All Proposed by IETF

# 综合服务模型（IntServ）概述

- ✓ **IETF的IntServ工作组于1994年提出**
- ✓ **同时支持实时和非实时业务**
  - 在每个实时业务开始之前，由终端设备向网络发出请求，为它预留必要的网络资源（带宽、存储），通过在网络中使用拥塞控制和队列调度机制，使实时业务的带宽和延迟得到保证
- ✓ **“流”：来自单一用户具有相同QoS需求，且可识别的数据分组**
  - **IntServ中能识别的最小粒度**
  - **流是单向的，它具有单一的发送端和N个接收端。**

# 服务类型

- ✓ 综合服务模型规定了三种不同等级的服务类型：
- ✓ (1) 有保证服务 (**Guaranteed-Service**, 简称**GS**)
  - 对带宽和端到端延迟有严格的界定, 并且保证数据流中合法的数据包无排队丢失。
- ✓ (2) 负载可控服务(**Controlled-Load-Service**, 简称**CLS**)
  - 保证在网络负载较重时能提供与负载较轻时相同的**QoS**保证。
  - 它与传统的因特网服务的主要区别在于它的性能不会随网络负载的加大而下降。
  - 负载可控服务一般用于可容忍一定的数据包丢失和延迟的应用。负载可控服务可由路由器中的队列机制实现。
- ✓ (3) 尽力而为服务(**Best-Effort-Service**)
  - 类似当前**Internet**在多种负载环境(由轻到重)下提供的尽力而为的业务, 目前不提供任何的**QoS**保证。

# IntServ中四个功能部件

- ✓ **资源预留协议**：负责逐跳（**hop-by-hop**）建立或拆除每个流的资源预留软状态（**soft state**），软状态是由资源预留周期性控制的临时状态，它包括有关该业务流的起止地址、路由信息，需占用该路由器的资源信息等。
- ✓ **接纳控制**：根据链路和网络节点的资源使用情况以及具体的**QoS**请求决定是否接受一个资源预留请求
- ✓ **分组分类器**：对到达的数据分组进行分类，然后分别放入不同的输出队列
- ✓ **队列调度器**：根据不同的策略对各个队列中的分组进行调度转发

# 业务量控制机制：接纳控制

- ✓ 根据当前资源情况，判断是否同意接入一个新的流的QoS请求
- ✓ 两种接纳控制算法
  - 基于资源预留参数的接纳控制：根据节点以前同意接入的所有请求的服务参数，用每个服务最坏情况的边界值来进行计算
  - 基于资源实际使用情况的接纳控制：测量现有分组流的实际链路带宽利用率，据此来判定是否接纳一个新的流

# 业务量控制机制：分组分类

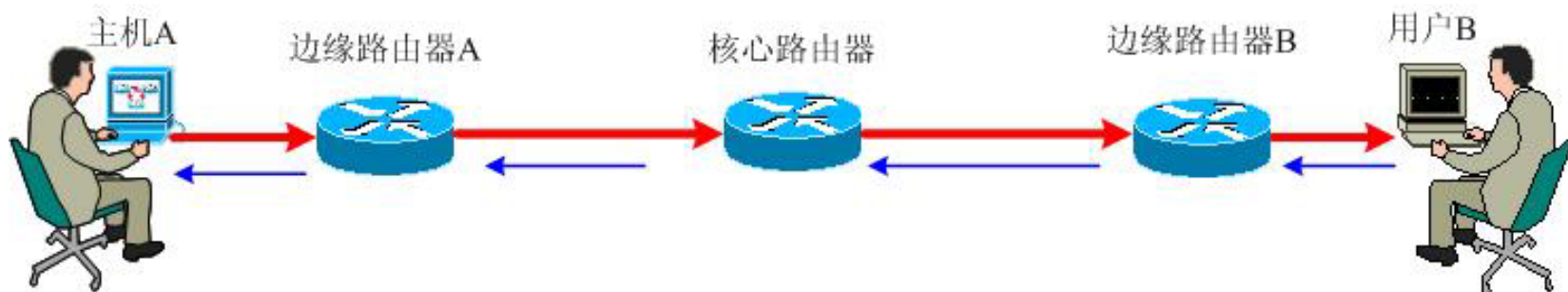
- ✓ 将输入分组映射到某个类，放入不同的输出队列
  - 一个类可能包含有多个流
- ✓ 多字段（**MF: Multi-Field**）分类器
  - 源/目的IP地址、源/目的端口、协议类型
- ✓ 原则：考虑分类粒度和处理开销之间的平衡
  - 用于分类的字段越多，处理开销越大

# 业务量控制机制：队列调度

- ✓ 从多个输出队列中选择下一个要转发的分组
- ✓ 分组调度算法
  - 严格优先级调度（**Strict Priority Scheduling**）
    - 队列按优先级递减排序，只有在高优先级的队列为空时，才服务下一个优先级的队列
    - 低优先级队列的分组可能被完全阻止发送（处于饥饿状态）
      - 映射到高优先权队列的业务量不超出允许的局部输出链路的容量
  - 加权公平队列**WFQ**（**Weighted Fair Queuing**）
    - 根据权重来确定每个队列所获得的链路带宽的百分比
      - **FQ**:  $F_i = \text{Max}(F_{i-1}, A_i) + P_i$

# 综合服务的结构和工作原理

- 假设主机A要向主机B传送有QoS要求的数据，数据从主机A发出，途经边缘路由器A、核心路由器和边缘路由器B到达主机B
  - 核心路由器(core router): 两个路由器之间传送数据的路由器
  - 边缘路由器(edge router): 将客户机连接到互联网的路由器
- 在主机A向主机B发送数据之前，主机A首先要与主机B建立联系，并请求沿途的路由器和其他交换设备保留资源，以建立保障服务质量的发送通道，然后开始传送数据，结束后通知沿途设备释放资源

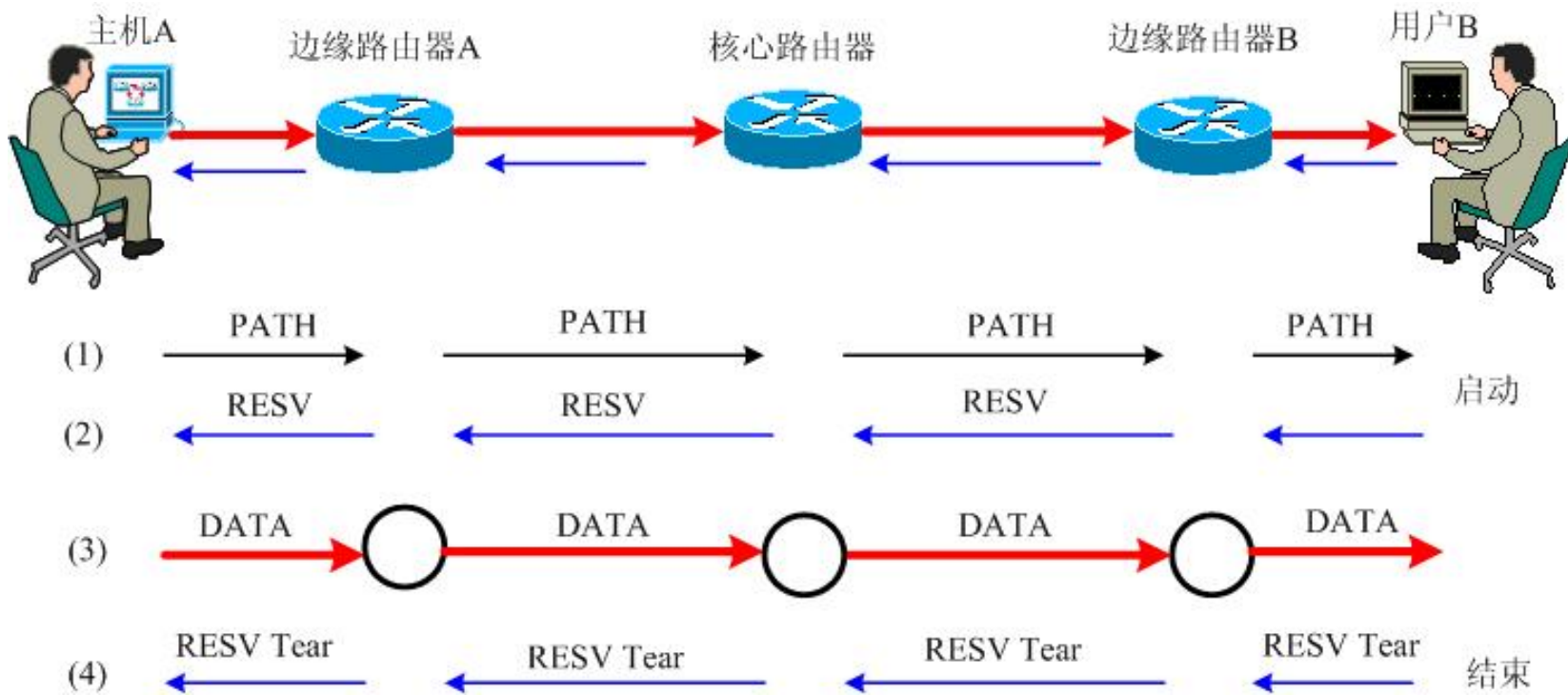




# 主要步骤

## ➤ 工作过程归纳成如下：

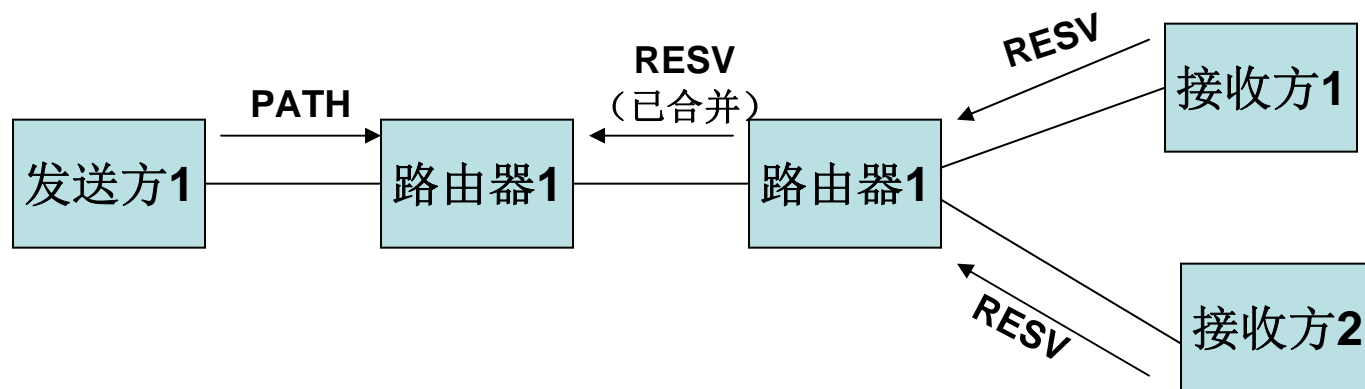
- 在发送实时数据前，主机**A**中的应用程序使用**RSVP**向主机**B**发送“**路径(PATH)**”消息，内含**QoS**要求。沿途的路由器和转发设备使用**PATH**消息搜索可用资源以**建立路径状态**
- 当主机**B**收到**PATH**消息后回送“**保留(RESV)**”消息，内含**实际的QoS**描述，沿途所有设备就可使用**QoS**描述建立路径状态，并意识到从源端到接收端有特定要求的数据包
- 当发送端主机**A**收到**RESV**消息后，如果**RESV**消息中没有错误消息，就向沿途要使用的所有设备发送**确认(confirmation)**消息，然后就开始发送数据包，沿途的设备就按照**QoS**的要求将数据包转发到接收端的主机**B**
- 数据传输结束后，发送端主机**A**发送一条结束传输的**PATH Tear**消息，告诉沿途的设备释放资源



## 综合服务的结构和工作原理

# 资源预留协议：特点

- ✓ 面向接收（**Receiver-Oriented**）：由接收方根据需求预留
- ✓ 软状态（**soft state**）：定期发送**PATH**和**RESV**消息维护
- ✓ 组播支持



# IntServ特点

- ✓ 提供端到端的**QoS**保证
- ✓ 基于流的细粒度资源分配
- ✓ 存在可扩展性问题
  - 必须建立和维护“每流”的预留状态信息
  - 需要对每个流进行接纳控制、分类等操作

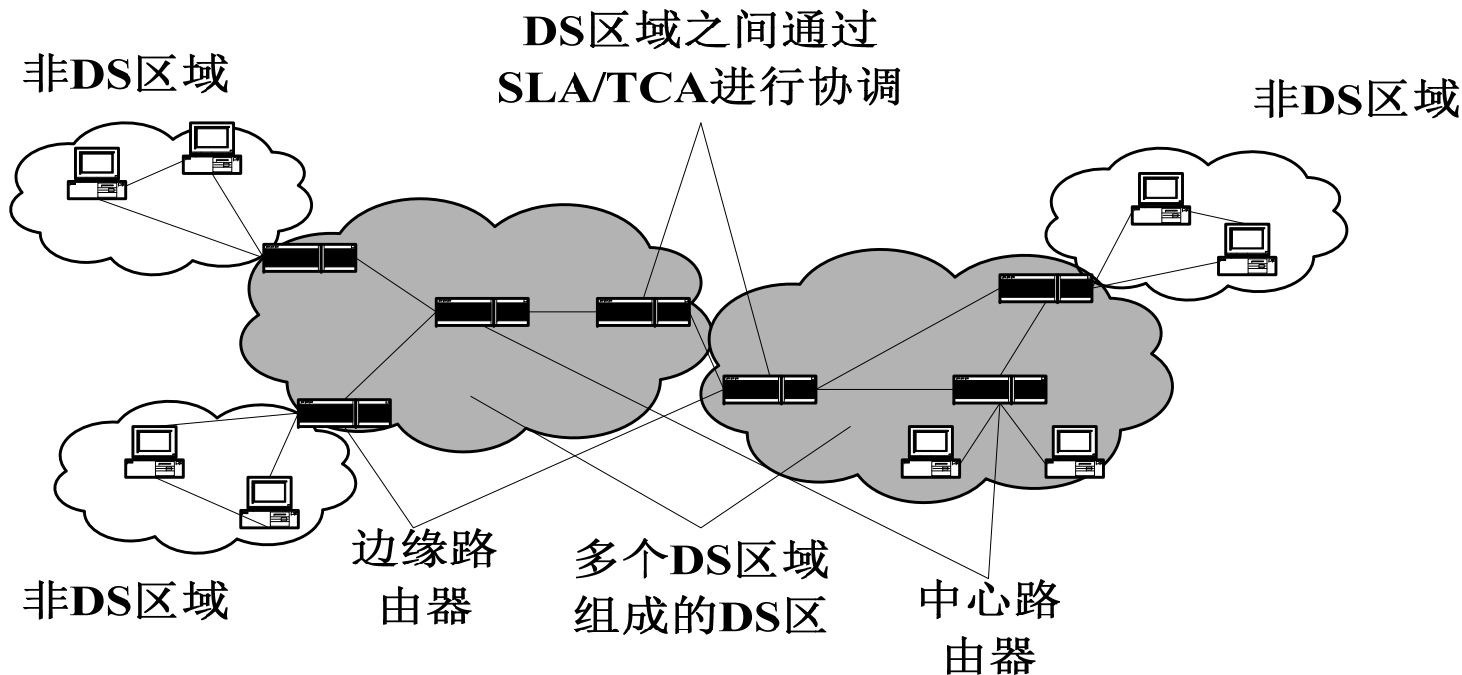
存储、处理开销随着流的数量增加而急剧增长

- ✓ 如果存在不支持**IntServ**的节点 / 网络，虽然信令可以透明通过，但对应用来说，已经无法实现真正意义上的资源预留，所希望达到的**QoS**保证也就打了折扣。

# 区分服务模型（DiffServ）概述

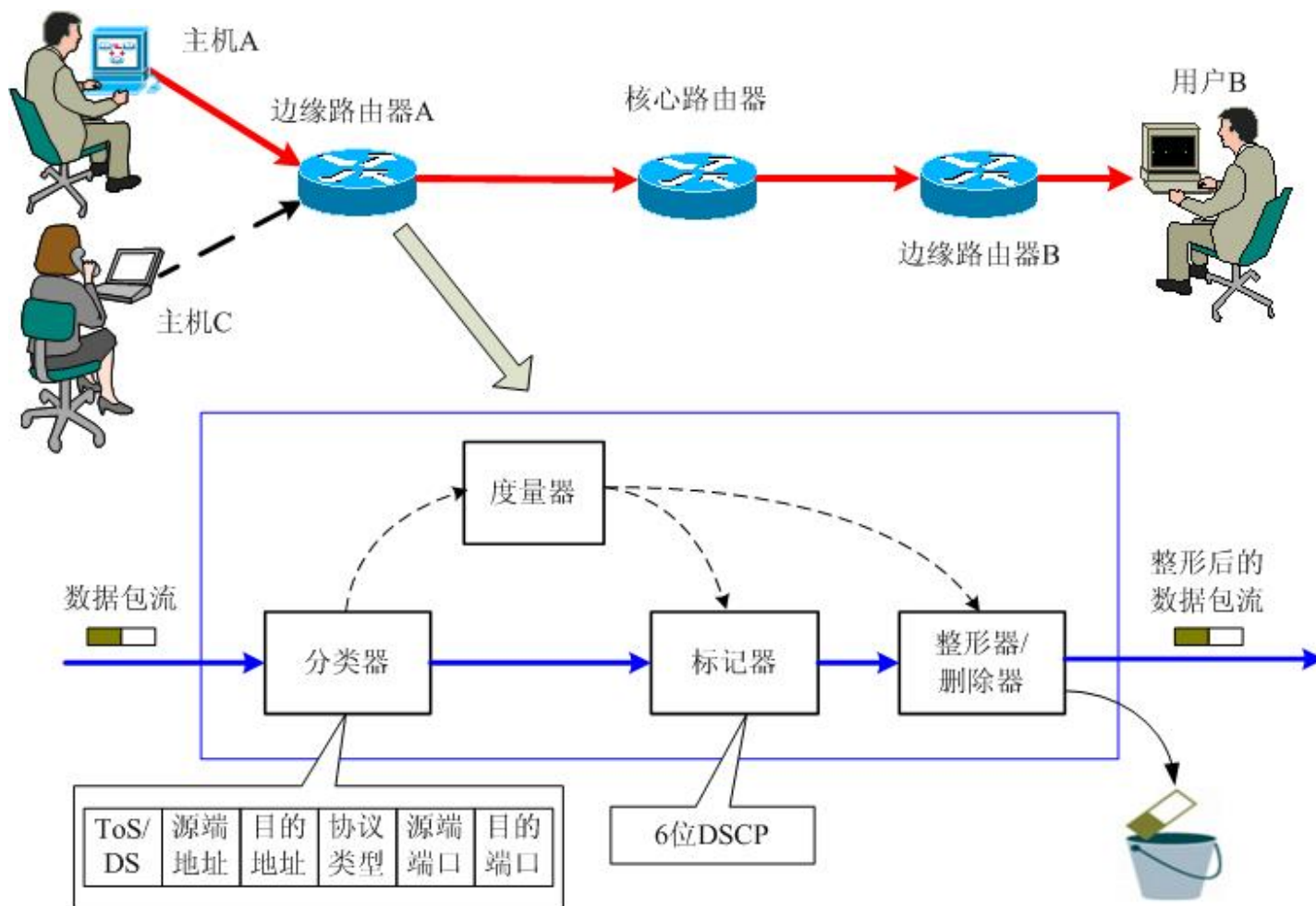
- ✓ IETF的DiffServ工作组于1998年提出
- ✓ 解决IntServ的可扩展性问题，在分组中携带的信息决定如何处理，而不需要使用RSVP协议
- ✓ 核心路由器存储转发的任务很重，核心网内不考虑复杂的QoS机制，而边缘路由器的流量相对较小，边缘路由器执行大部分的QoS实现机制。
- ✓ 基于类的QoS保证，通常在核心网中使用
  - 当业务到达区分服务区域的边界时，边缘路由器使用分组头标中的区分服务标记域（DS field）对其进行聚类，网络对同类业务给予相同的QoS保证

# 结构模型 (1)

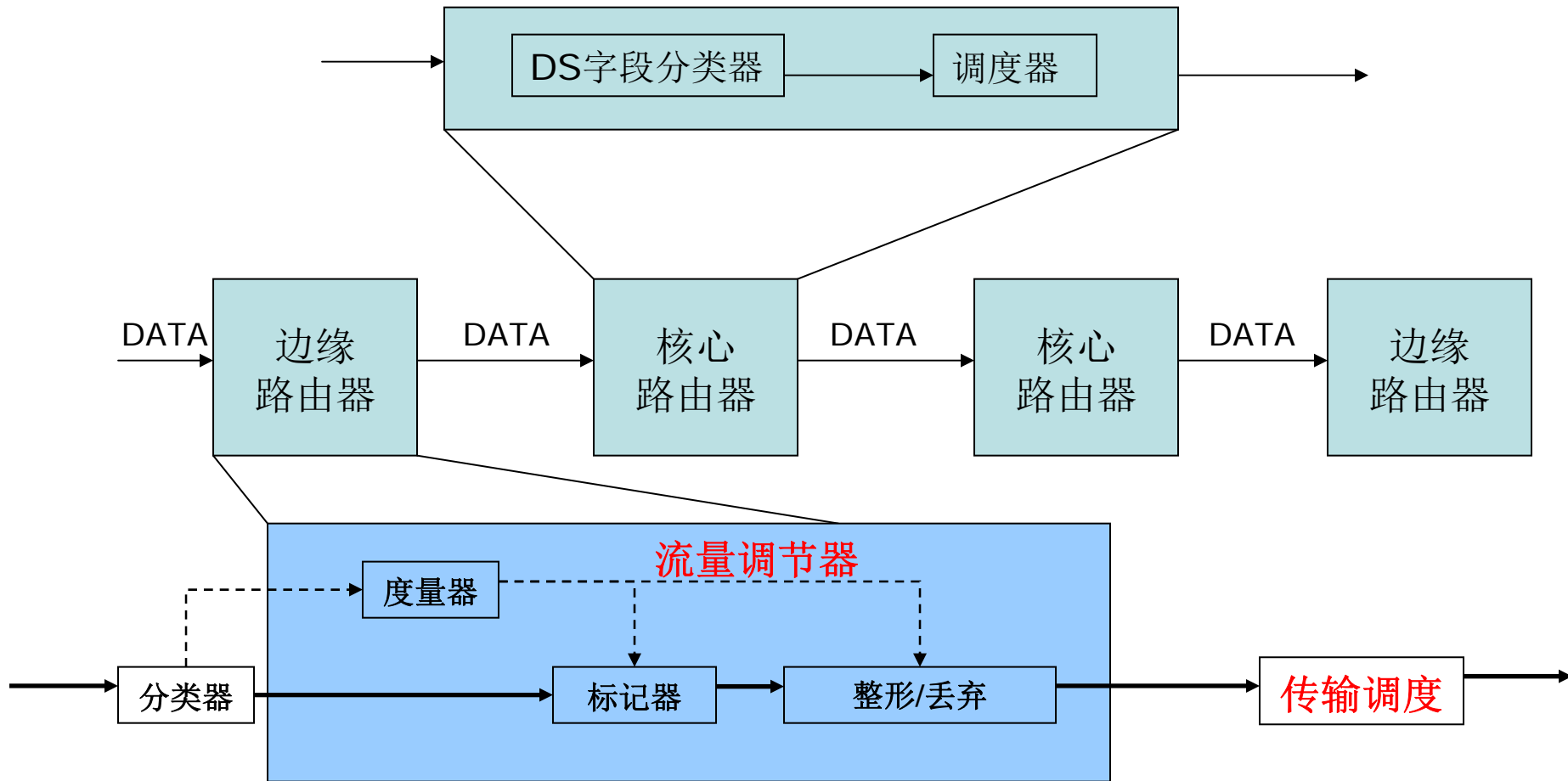


- ✓ 边缘路由器：业务量分类和调节，对分组头标中的**DS**域进行标记，标记值被称为**DSCP**
- ✓ 中心路由器：根据IP分组**DS**域中所标记的**DSCP**值，来选择所对应的转发处理，即逐跳行为（**PHB**），从而对分组进行调度转发
- ✓ **SLA**（**Service Level Agreement**）协商：不同**DS**区域之间的分类规则、重新标记规则以及业务流应该符合的业务量配置文件
- ✓ **TCA**：流量调整协定，关于业务分类准则、业务模型及相应处理的协定。

# 区分服务在边缘路由器上的概念模型



# 结构模型 (2)

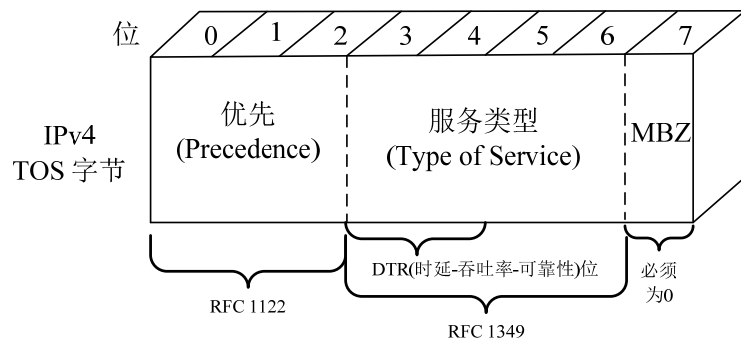




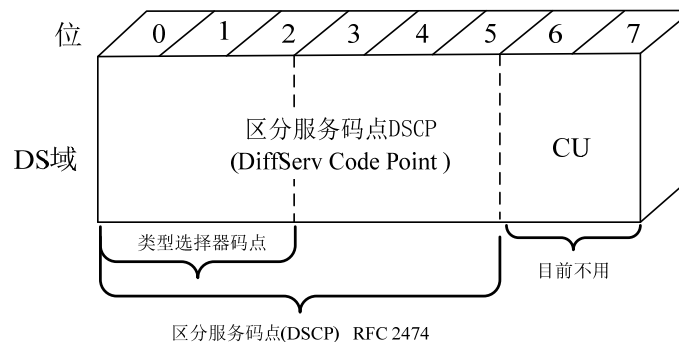
# DS域定义与DSCP

## ✓ DS域定义

- IPv4头标的TOS（Type of Service, TOS）域或IPv6头标的业务量等级（Traffic Class）域的前6比特



(a) 原IPv4 ToS字节



(b) 区分服务码点域

- ✓ DSCP: DS域中的具体值，DS节点根据DSCP选择特定的PHB

# 逐跳行为（PHB）

- ✓ 服务类型使用逐跳行为(**per-hop behavior, PHB**)定义
  - **hop**: 路由段, 通常是指两个路由器之间的链路。在互联网上, 大多数数据包的路径由多个路由段组成
  - 行为(**behavior**): 转发数据包的行为, 包括对数据包的调度、排队和整形等
  - 逐跳行为(**PHB**)根据服务等级协议(**SLA**)进行设置。不同的**PHB**有不同的时延、抖动、丢包率和吞吐率的限制, 因此需要有不同的转发行为
- ✓ 段行为(**PHB**)使用**6位**区分服务码点(**DSCP**)表示
- ✓ **PHB**描述了根据**DSCP**对特定类进行资源分配的方式
  - 具体由相应的队列调度算法例如**WFQ**来实现
- ✓ 四种**PHB**
  - **BE (Best Effort)**, 尽力而为型, 缺省型**PHB**
  - **EF (Expedited Forwarding)**, 加速转发型**PHB**
  - **AF (Assured Forwarding)**, 可靠转发型**PHB**
  - **CS (Class Selector)**, 兼容**IP**优先级的类选择型**PHB**

# EF PHB: 加速转发型PHB

✓ 为某种业务量集合提供低丢失率、低延时和低延时抖动的服务，最高服务优先级

✓ 需要以下两部分功能模块：

流量调节器完成, 丢弃超过预定带宽的业务量

➤ 调节这个集合的业务量（通过监控和整形）使它在任意节点上的到达速率总是小于那个节点配置的最小离开速率。

➤ 配置节点，使得某个集合的业务量有一个独立于节点其他业务量的离开速率（配置速率）。

✓ 推荐DSCP为101110

调度算法(WFQ)来实现

# AF PHB:可靠转发型PHB

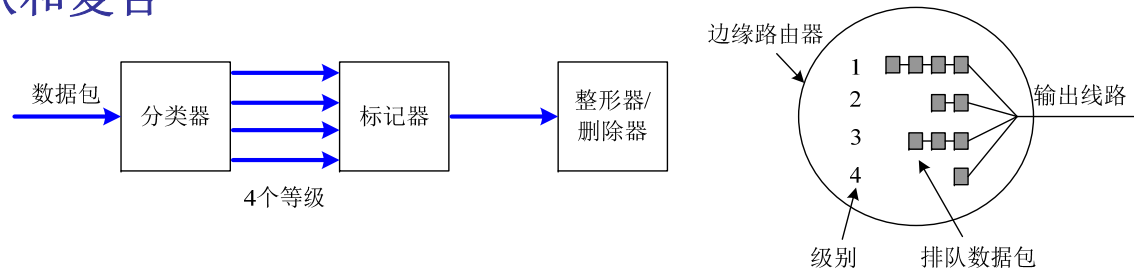
- ✓ 为用户提供不同级别的转发保证,定义了四个等级,为每个等级都分配特定数量的转发资源
- ✓ IP分组转发保证取决于:
  - 多少资源分配给此分组所属的AF等级
  - 此AF等级当前的负载和拥塞情况
  - 分组的丢弃优先级
- ✓ DSCP

服务等级 丢弃优先级	等级 1	等级 2	等级 3	等级 4
低	001010	010010	011010	100010
中	001100	010100	011100	100100
高	001110	010110	011110	100110

# AF PHB:可靠转发型PHB

可靠转发数据包的步骤如下

- 使用分类器(classifier)将数据包分到4个等级中的一个
- 按照给定的服务等级，使用标记器(marker)对数据包做标记，也就是分配区分服务码点(DSCP)
- 将数据包送到整形器/删除器(shaper/dropper)进行调整，产生符合服务质量等级要求的数据包流。其中
  - » 整形器用于“交通整形(traffic shaping)”，目的是将速率不同的输入数据包流变成输出速率恒定的数据包流
  - » 删除器用于“交通维持(traffic policing)”，检查数据包流中是否有不符合服务等级要求的数据包，如果不合就请它出去
- 在边缘路由器中排队和复合
- 发送到输出线路上



# 兼容IP优先级的类选择型PHB: CS PHB

- ✓ 向后兼容IP优先级队列,历史上IPv4 TOS域的前3比特曾作为优先级队列调度的标志,共定义了8个优先级
- ✓ 可通过严格优先级、循环优先级队列等较为粗略的调度机制实现
- ✓ **DSCP: xxx000**

# DSCP值

名 字	DSCP(十六进制) ( CU=00)	DSCP(二进制) ( CU=00)
<b>Default</b>	<b>0</b>	<b>000000 00</b>
<b>AF11</b>	<b>28</b>	<b>001010 00</b>
<b>AF12</b>	<b>30</b>	<b>001100 00</b>
<b>AF13</b>	<b>38</b>	<b>001110 00</b>
<b>AF21</b>	<b>48</b>	<b>010010 00</b>
<b>AF22</b>	<b>50</b>	<b>010100 00</b>
<b>AF23</b>	<b>58</b>	<b>010110 00</b>
<b>AF31</b>	<b>68</b>	<b>011010 00</b>
<b>AF32</b>	<b>70</b>	<b>011100 00</b>
<b>AF33</b>	<b>78</b>	<b>011110 00</b>
<b>AF41</b>	<b>88</b>	<b>1000010 00</b>
<b>AF42</b>	<b>90</b>	<b>100100 00</b>
<b>AF43</b>	<b>98</b>	<b>100110 00</b>
<b>EF</b>	<b>B8</b>	<b>101110 00</b>

# IP 优先权

优 先 级	IP 优先级(十六进制) (TOS=000, ECN=00)	IP 优先级(二进制) (TOS=000, ECN=00)
0	00	000 000 00
1	20	001 000 00
2	40	010 000 00
3	60	011 000 00
4	80	100 000 00
5	A0	101 000 00
6	C0	110 000 00
7	E0	111 000 00



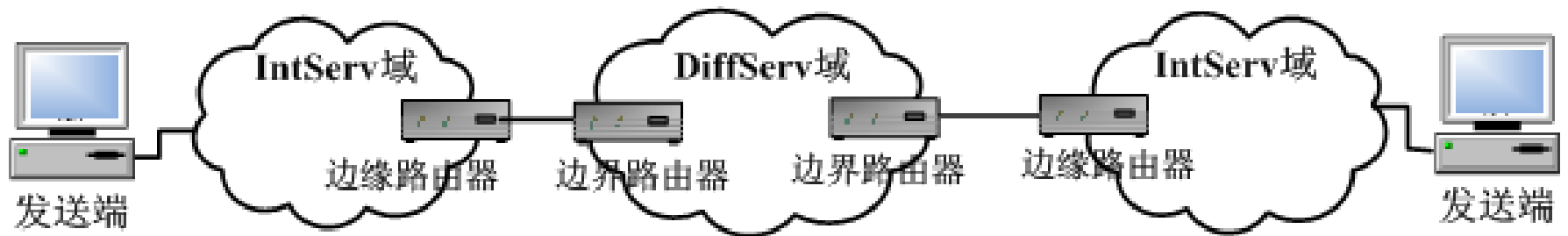
# 区分服务的工作过程归纳

- ✓ 在边缘路由器(如A)上，
  - 按照IP数据包头域中的源端地址、目的地址、源端口号、目的端口号和区分服务类型(**ToS/DS(Type-of-Service / Differentiated Service)**),
  - 使用分类器(**classifier**)对输入数据包进行**分类**,
  - 然后使用等级度量器(**meter**)和等级标记器(**marker**)对数据包进行标记(**6位DSCP**),
  - 使用等级度量器和交通整形器/删除器(**shaper/dropper**)对交通进行整形和调整, 在边缘路由器上排队后, 最终生成符合服务质量等级要求的数据包流。
- ✓ 在核心网络(**core network**)中, 网络设备按照用码点标识的“**段行为(PHB)**”转发数据包

# DiffServ特点

- ✓ 基于聚合类的粗粒度资源分配
- ✓ 具有较好的可扩展性。**DS**字段只是规定了有限数量的业务级别，状态信息的数量正比于业务级别，而不是流的数量
- ✓ 易于实现。只在网络的边界上才需要复杂的分类、标记、整形等操作
- ✓ 无法提供端到端的**QoS**保证

# Diff-IntServ综合模型



- ✓ IntServ模型：接入网络
- ✓ DiffSev模型：核心网络
- ✓ IntServ和DiffServ网络之间的映射机制

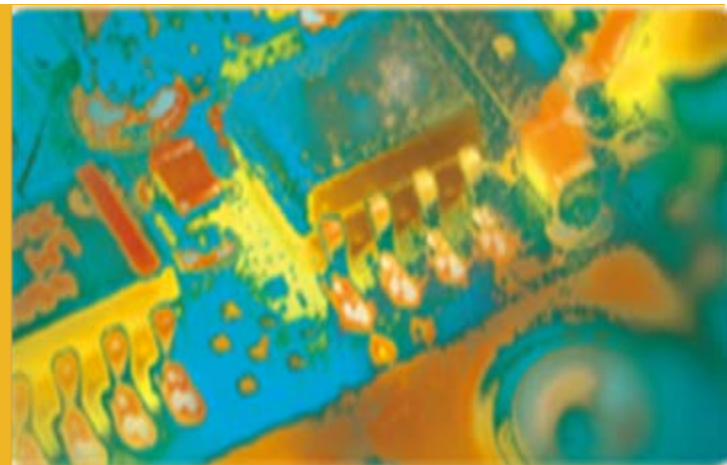


端到端QoS



温州大学  
WENZHOU UNIVERSITY

# 第7章 网络服务质量 测试



## 7.4 QoS中的队列调度与管理技术

# 7.4.1 队列调度与管理概述

- ✓ 拥塞产生的直接原因有如下三点：
  - 存储空间不足：没有足够的存储空间，数据包则会被丢弃；但只一味的增加存储，数据包可能会排队超时而丢弃；
  - 带宽容量不足；
  - 处理器处理能力弱、速度慢。如果路由器的 **CPU** 在执行排队缓存、更新路由表等功能时，处理速度跟不上高速链路，也会产生拥塞。
- ✓ 一直在关注、研究着拥塞控制算法

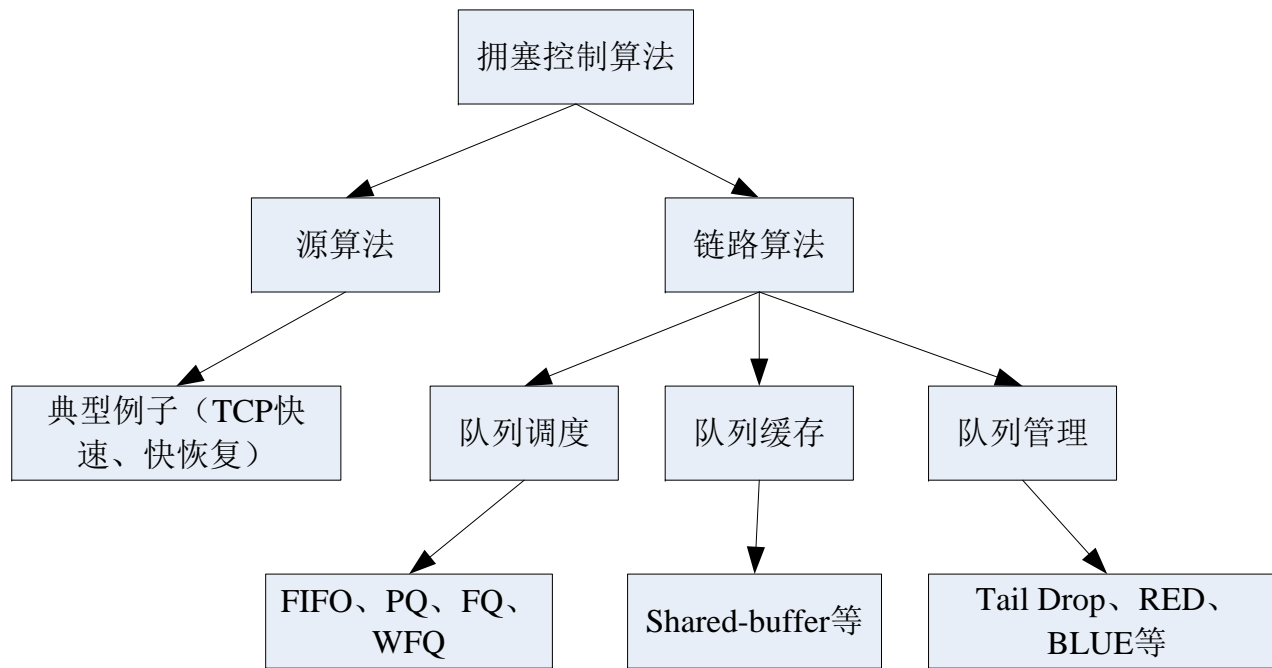
# 拥塞控制算法分为两大类

## ✓ 源算法。

- 指在端系统(主机和网络边缘设备)中运行的算法
- 作用是根据反馈信息，调整发送速率
- 典型代表有**TCP**的拥塞控制算法，包括慢开始、快重传和快恢复等技术；

## ✓ 链路算法

- 在网络中间设备（如路由器和交换机）上实施并执行
- 作用是检测到网络拥塞的发生，采取一定的措施缓减拥塞状况，产生拥塞反馈信息以使发送端采取适当的措施避免拥塞恶化。



- ✓ 队列调度机制又称为排队和调度机制。
  - 输入流量按照分类送入不同的队列排队，
  - 队列调度对不同优先级的流量按一定的调度策略进行处理，优先级高的流量会得到优先处理。
- ✓ 队列缓存管理的任务是为不同数据流分配缓存空间。
  - 包括各种动态或静态策略
  - 共享缓存池（**shared buffer pool**），共享缓存池以先到先服务为基础，其优点是简单和执行效率高，缺点是缺乏数据流之间的保护
  - 单一流分配（**per-flow allocation**），单一流分配为每一个数据流建立一个缓存使用记录，根据每一个流的缓存占用水平决定分组丢弃，从而起到保护各个流的作用。但是这样的处理过程太耗费资源，因而无法在骨干路由器上大规模的推广。



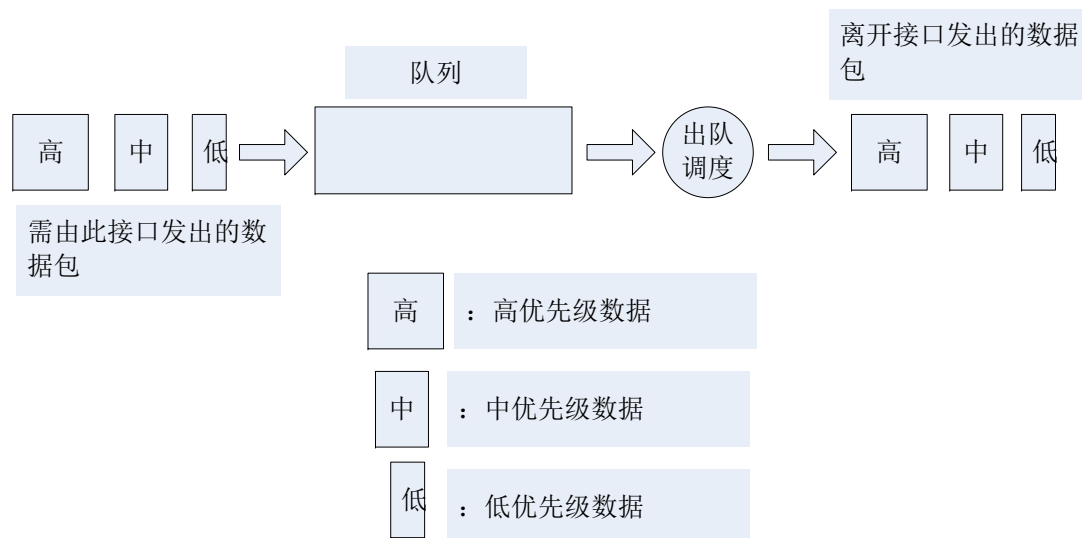
- ✓ 队列管理的任务是通过选择丢弃分组以及丢弃时机来控制队列的长度和调度数据流。
- ✓ 队列管理是调度机制和缓存管理的补充，同时也是二者的交叉结合。
- ✓ 目标是保证链路可用时队列长度处于稳定状态。
- ✓ 主动队列管理是当前队列管理的热点，**RED**算法是当前使用最为广泛的主动队列管理算法。

- ✓ 为了保证**QoS**,
- ✓ 运行队列调度算法将把属于不同服务等级的分组送入不同的队列单元中进行排队
- ✓ 在拥塞时, 通过队列管理, 采用牺牲低优先级的分组的策略, 以保证高优先级分组的需求。

## 7.4.2 常见队列调度技术

### ✓ 先进先出排队 **FIFO** (First-in/First-out)

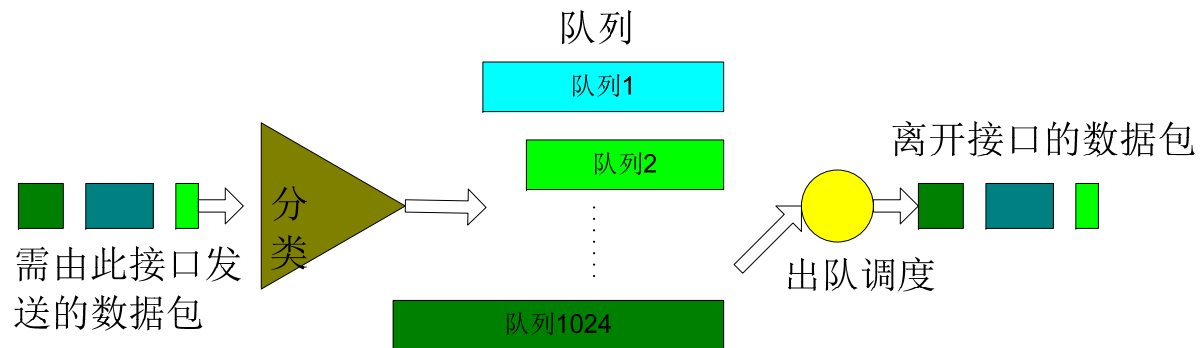
- 分组不管其优先级大小，都按到达的时间顺序放置于这个队列，按照分组在队列中的顺序依次转发
- 分组的丢弃采用了队尾丢弃的原则
- 优点：便于实施，缺点：其公平性较差（高速率的数据流占用更多带宽）



# 7.4.2 常见队列调度技术

## ✓ 公平队列FQ (Fair Queuing)

- 分组被分成不同的流，并被置于与特定流相对应的队列中。在这种调度机制中，循环发送多个队列中的分组。
- 优点：带宽分配独立于数据包大小，提供了较好的公平性，可以较好地与端到端的拥塞控制机制协同
- 它的缺点在于实现起来很复杂



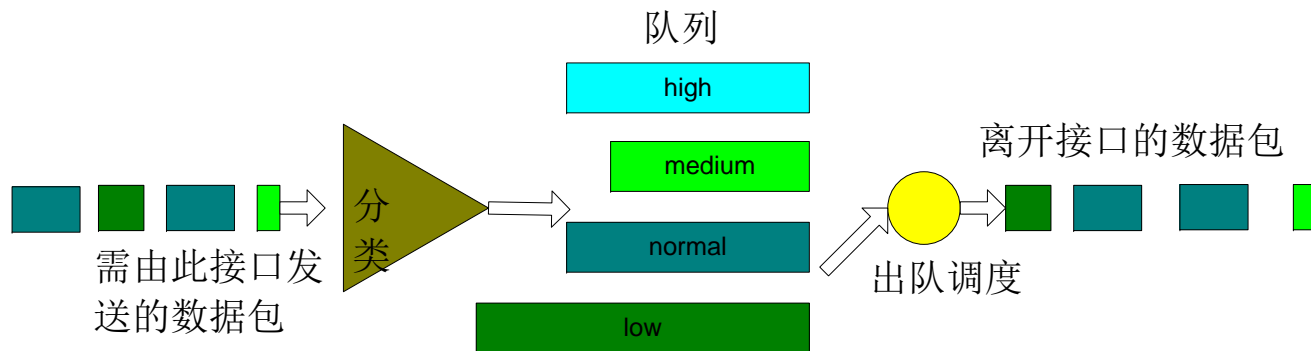
## 7.4.2 常见队列调度技术

- ✓ 加权公平队列**WFQ**（**Weight Fair Queuing**）
  - **FQ**的改进算法
  - 系统根据待发送数据流所需的带宽为每一个队列分配一个输出带宽百分比，权值大的队列将得到更多的带宽份额。
  - **WFQ**通过区分不同长度的分组，能够防止含有长分组的待发数据流比短分组数据流获得更多的带宽所导致的不公平问题。
  - 图7.7

## 7.4.2 常见队列调度技术

### ✓ 基于优先级的队列PQ (Priority Queuing)

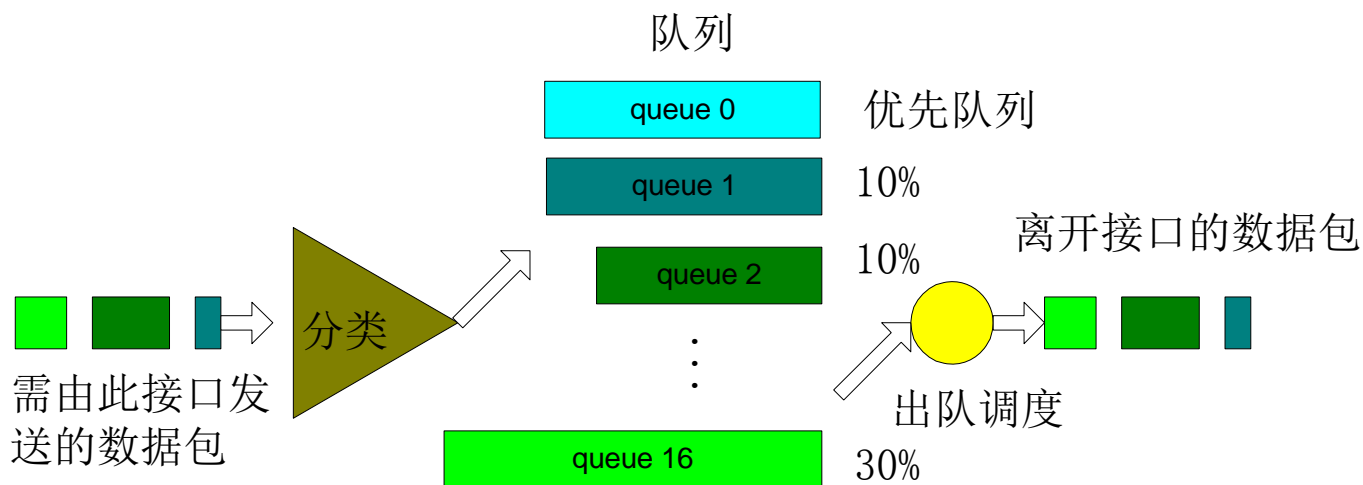
- 分组首先被分成不同的优先级类别，并被置于具有不同优先级的队列中。 队列长度可配置
- 位于高优先级队列中的分组将首先获得服务，直到高优先级队列中没有等待发送分组，低优先级队列才可能获得被服务的机会
- **FQ**可能导致的一个问题是饿死



## 7.4.2 常见队列调度技术

### ✓ 基于自定义的队列CQ（Custom Queuing）

- 为不同协议分配不同的队列空间，可配置队列间占用的带宽比例关系，并以循环方式处理队列
- 当特定协议的数据流被分配了较大的队列空间，也就获得了较优先的服务



## 7.4.3 主动队列管理技术

- ✓ 网络的流量具有突发性、自相似性，因而给网络的路由节点造成很大的负担，使得拥塞情况不断发生，假定大多数在数据传输应用中进行的通信对丢失很敏感。在Internet中避免高的包丢失率是非常重要的。
- ✓ 传统尾丢弃方式造成TCP全局同步，使线路上的流量总在极少和饱满之间波动。
- ✓ TCP拥塞机制只能提供有限的避免拥塞能力

震荡吞吐量

无RED





# 什么是TCP全局同步

- 由于内存资源的有限，传统的拥塞处理方法是，当队列的长度达到规定的最大长度时，所有到来的报文都被丢弃。
- 大量的TCP报文被丢弃，将造成TCP超时，引发TCP的慢启动和拥塞避免机制，使TCP减少报文的发送。
- 当队列同时丢弃多个TCP连接的报文时，将造成多个TCP连接同时进入慢启动和拥塞避免，称为“TCP全局同步”：多个TCP连接发向队列的报文将同时减少，
- 回避一段时间后，又同时增加发送流量，使得网络上的流量忽大忽小，网络上的流量总在极少和饱满之间波动。
- 降低了线路带宽的利用率。

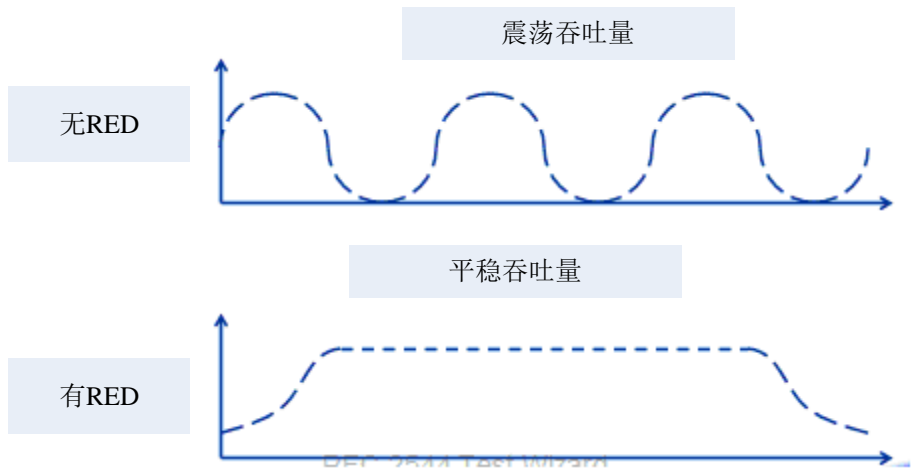
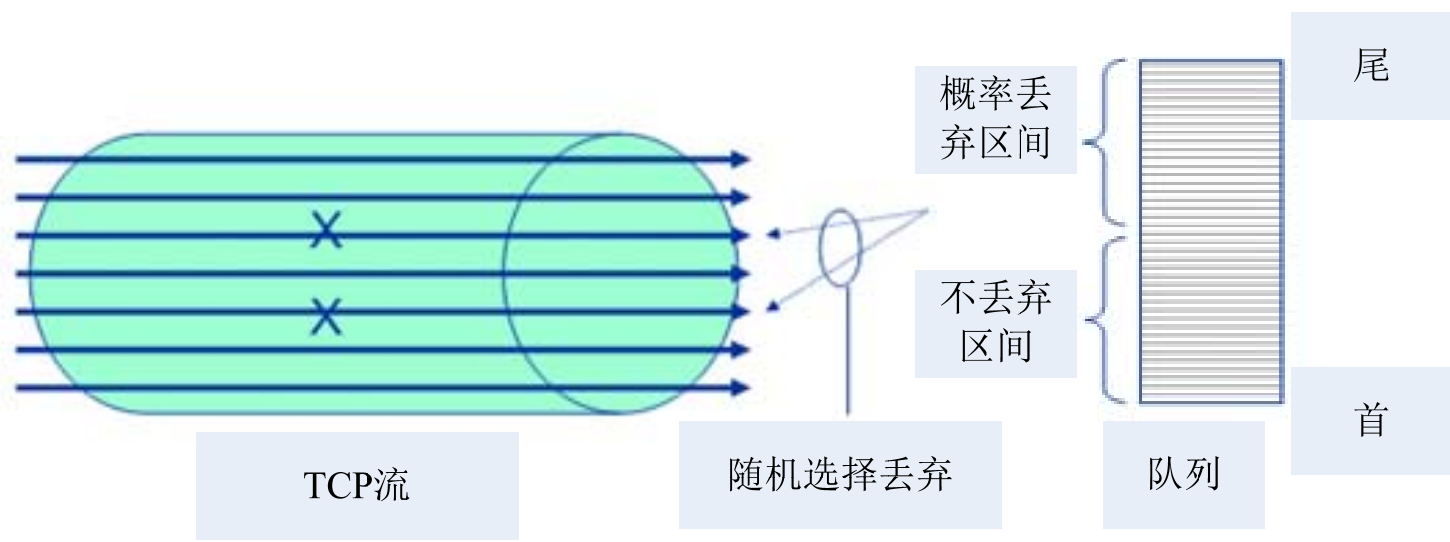
## 7.4.3 主动队列管理技术

- ✓ 如何才能少路由器中丢弃的包的数量？
- ✓ 主动式队列管理**AQM**（**Active Queue Management**）：在路由器队列充满之前丢包的方式，使端节点能在队列溢出前对拥塞做出反应
- ✓ 采用随机丢弃的方式，避免了传统尾丢弃方式引起的**TCP**全局同步现象发生。提高了线路带宽的利用率。

## 7.4.3 主动队列管理技术

- ✓ 用户可以设定队列的低限和高限。
- ✓ 当队列的长度小于低限时，不丢弃报文。
- ✓ 当队列的长度在低限和高限之间时，**WRED**开始随机丢弃数据包。并且，队列的长度越长，丢弃的概率越高。
- ✓ 当队列的长度大于高限时，丢弃所有的数据包。

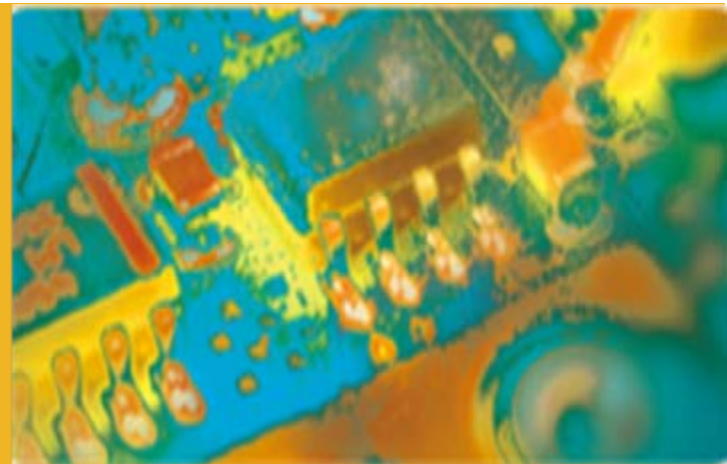
# 7.4.3 主动队列管理技术





温州大学  
WENZHOU UNIVERSITY

# 第7章 网络服务质量 测试



## 7.5 网络服务质量测试方法学

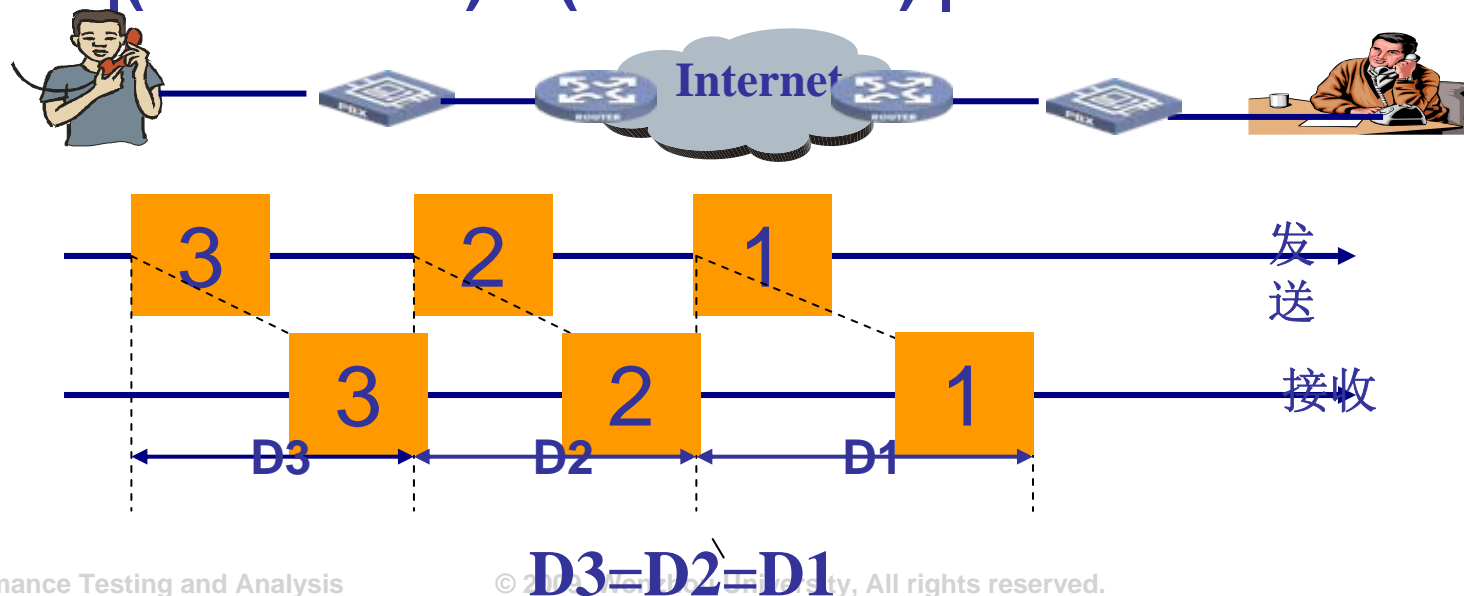
# 7.5.1 QoS性能参数

✓ 主要是网络的5个传输特性参数：时延、抖动、丢包率、错序、吞吐率和服务可用性

## ✓ 1. 时延抖动

➤ 时延抖动是指沿同一路径传输的数据流，不同数据包传输的时延变化。

➤ 
$$\text{jitter} = |(RxA - TxA) - (RxB - TxB)|$$



# 7.5.1 QoS性能参数

## ✓ 2. 错序

- 错序是指发生了接收端接收到的数据包排列顺序与发送端不同的现象。
- 对ftp、http等影响不大，但对IPTV、IP电话等实时业务却会产生严重的影响

## ✓ 3. 服务可用性(service availability): 用户连接互联网时获取网络资源的难易程度。定义在给定的时间范围里，网络可提供的服务时间占给定时间间隔的百分比

## 7.5.2 测试方法和内容

- ✓ 1. 明确被测对象所部署的**QoS** 策略
  - 如服务区分点、服务等级的区分等，并且确认测试目的。
- ✓ 2. 确定测试方法
  - 必须在**流量拥塞**的情况下，测试被测对象部署的**QoS**策略
  - 制造拥塞方法：通常采用输入带宽大于输出带宽这种非对称的方式设计测试流量
    - 多端口向一个端口发送流量
    - 双端口采用非对称带宽



## 7.5.2 测试方法和内容

### ✓ 3. 选定性能指标和性能评价标准

#### ➤ 考虑3个方面

- 从网络的整体效率性能：用网络的吞吐率表示
- 满足用户应用要求：采用延迟和丢失率描述
- 公平性：比较复杂，没有合适的标准
  - 如网络资源的平均分配，
  - 如网络拥塞时，网络资源合理分配的策略

#### ➤ 资源有限，难以同时满足，有时相互限制，采用折中组合，如：

- 吞吐率+延迟
- 延迟+公平性、丢失率+公平性

# 1. 时延抖动测试方法

## ✓ 1) 基于时延的测试方法

## ✓ 2) 基于到达时间的测试方法

- 在测试过程中，测试源发送固定长度（例如**128byte**）报文，目的端记录每个报文的到达时间
- 测试持续时间至少**120**秒以上。
- 假设目的端先后到达了**A**、**B**、**C**三个数据包，其中 **A** 与**B**到达的时间差记为**T1**，**B**与**C**到达的时间差记为**T2**，那么**|T1-T2|**就是时延抖动值。
- 测试中由**N**个连续的测试数据包求出**N-1**个时延抖动值，取其平均值为平均时延抖动值。

## 2. 错序测试方法

### ✓ 发送端

- 为了标记每个数据包，在发送数据包中加入标识其位置的**序号**（**Sequence Number**）
- 用于跟踪期望收到数据包序号的**最迟阈值**，最迟阈值大小可以自定义，以固定长度，固定发送速率发送数据包；

### ✓ 接收端

- 用当前期望值表示期望收到的数据包序号，当前期望值由收到所有测试包中最大序号加**1**确定
- 如果收到的数据包的序号与所期望接收包的序号差值
  - **0**，无错序
  - 小于最迟阈值，那么就认为该包是因为时延而错序，
  - 大于最迟阈值则认为是不可以重新正确排序的错序。
- 由于错序的测试是一个随机参数过程，该测试必须反复进行，所得到的报告值应该取记录的平均值。

# 实例

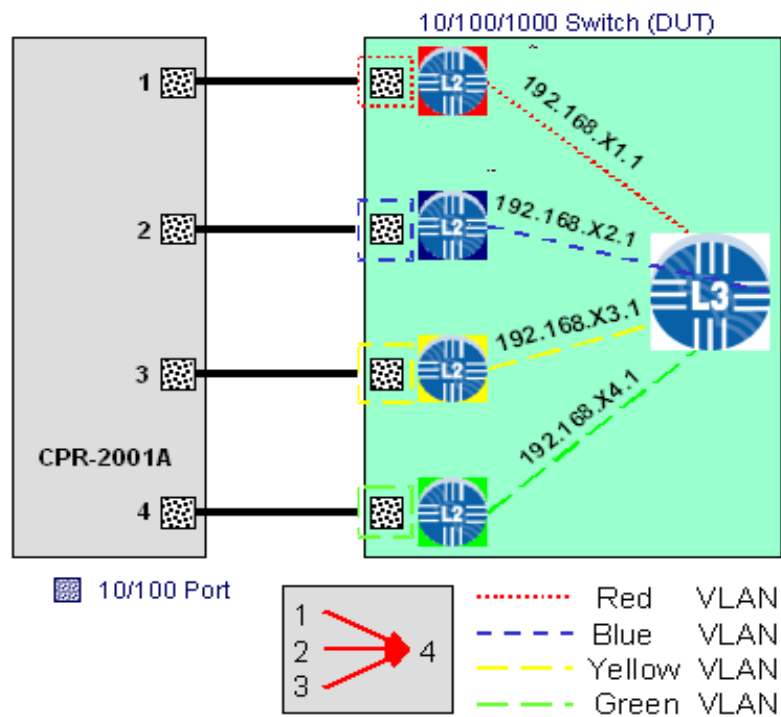
- ✓ 假设测试源发送序号为**1~25**的数据包，最迟阈值为**5**。目的端收到的数据包次序为：**2、3、5、4、6、10、7、8、9、11、12、13、14、1、11、15、25、16、17、18、19、20、21**。
- ✓ 在目的端接收数据包的过程中，收到**5**号数据包后，当前期望值为**6**，接着收到**4**号数据包，由于序号**4**小于**6**，所以**4**号数据包不是所期望的数据包，但是序号**4 > 1**（期望值**6**减去最迟阈值**5**），因而**4**号数据包是可以重新排序的错序。
- ✓ 同理，序号为**7、8、9**的数据包也是可以重新排序的错序包。收到**14**号数据包后，当前期望值为**15**，接着收到**1**号数据包，由于序号**1**小于**15**，所以**1**号数据包不是所期望的数据包，但是序号**1 < 10**（期望值**15**减去最迟门限值**5**），因此**1**号包是不可重新排序的错序数据包。

# 实验基于DSCP的三层区分服务测试

## ✓ 实验目的

- 进一步理解区分服务概念
- 掌握依据区分服务**DSCP**字段配置优先级的方法
- 掌握基于不同区分服务优先级创建测试流量的方法
- 通过对测试结果的分析，进一步理解区分服务处理流程
- 掌握通过结果报告器（**Results Reporter**）生成**PDF**测试报告的方法

## ✓ 实验环境与拓扑



Logical Map

## ✓ 实验内容

- 根据IP报文首部**DSCP**字段为**Spirent testcenter**的三个端口（端口**A**、端口**B**、端口**C**）分别配置高、中、低三个优先级的流量
- 将**3**条流量发送到第四个端口（端口**D**，接收端）
- 实验中逐步加大发送流量，故意造成接收端拥塞，
- 观察并分析接收端在拥塞情况下包处理的各种结果

## ✓ 配置对象

- 端口

- 主机

- 不同优先级的流量

- 流量负载大小