

## 一种改善 IP 网络中基于类的数据服务 QoS 的调度方法

郭小清 吴介一 张颀兵  
(东南大学 CIMS 中心 南京 210096)

**摘要** IP 网络中基于类的数据服务 QoS 支持一般包括包的延时、吞吐量和丢失率, 目前的研究成果大都集中于对其中某一个指标的确保或支持。该文提出一种在 IP 网络中一个网络路由器节点上同时考虑类间流量带宽、包丢失和延时区分的队列管理调度算法, 称为动态比例自适应算法(Dynamic Proportional Adaptive Algorithm, DPAA)。该算法是针对比例区分服务 PDS(Proportional Differentiated Services)中流量负载频繁变化的情况提出的。仿真实验通过与前人提出的基于比例区分服务模型的调度算法比较, 证明了 DPAA 算法的有效性。

**关键词** 比例区分服务, 调度算法, 服务质量

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2006)03-0557-05

## A Scheduling Algorithm for Improved QoS Support of Multiclass Data Service on IP Networks

Guo Xiao-qing Wu Jie-yi Zhang Sha-bing  
(The CIMS Center, Southeast University, Nanjing 210096, China)

**Abstract** QoS supports of multiclass data service in IP network includes generally packet delay, throughput and loss rate, while the majority last research focus on either minimizing packet delay and loss or maximizing user throughput. In this paper, an algorithm DPAA (Dynamic Proportional Adaptive Algorithm) for assuring the difference service under traffic vary-time condition in Proportional Differentiated Services(PDS) model is proposed. The scheme takes into account the throughput, delay and loss rate of packets between classes simultaneously. Simulation indicates that the DPAA has a good performance compared to the previous works.

**Key words** Proportional Differentiated Services, Scheduling algorithm, QoS

### 1 引言

在Internet中针对不同应用需求提供区分服务是ISP及研究者们近年来的一个热点课题, 尤其随着多媒体应用的普及, 传输对网络带宽、延时和丢失率等有较高敏感性的实时视频流是其中一个难点。现有Internet是基于TCP/IP的单一服务网络, 即对所有流提供尽力而为服务, 无法满足实时视频流的要求, 于是做为QoS研究成果的DiffServ<sup>[1,2]</sup>与IntServ<sup>[3]</sup>结合传输的方案<sup>[4]</sup>被广泛接纳。

但是DiffServ规范只定义了数据包的转发行为, 对怎样通过RSVP, 集成服务和自适应区分服务等机制构建一个可预测应用仍是一个开放的问题, 定位多媒体应用最优使用DiffServ网络资源的方法亦是一个挑战。Dovrolis等提出的PDS<sup>[5]</sup>是DiffServ框架的一种基本的实现方法。这个方法中没有接入控制和资源预留, 只是保证高级别类中的流会得到相对较好的服务质量, 即对于 $i \in [1, N]$ , 有 $\alpha_i/\alpha_{i+1} = \beta_i/\beta_{i+1}$ , 其中 $\alpha_i$ 表示类 $i$ 的服务区分参数,  $\beta_i$ 表示类 $i$ 的服务质量。

基于比例区分服务模型, 研究者们已经提出了一些保证服务质量区分的队列管理和调度机制。文献[5]针对延时区分

保证提出了WTP和PAD两种调度器, 随后在文献[6]中提出了保证丢失率的 $P(\infty)$ 和 $P(M)$ 机制。文献[7]综合考虑吞吐量和延时提出了一个混合调度管理机制, 寻找吞吐量和队列长度变化相对最小的类优先服务, 且证明无法在不知道实际丢失率的情况下同时保证丢失率服务。文献[8]通过实时统计每个类中的活动流数量动态调整队列调度权重和缓冲区分配保证基于流的带宽和延时比例区分服务。这些机制都是独立考虑带宽、延时、丢失区分或其中的两个。

本文基于比例区分服务模型, 同时考虑带宽、延时和包丢失率因素提出一个队列管理调度机制DPAA, 这个机制主要针对网络负载较大的情况。通过类RED的方式进行队列管理, 动态调整每个类的缓冲区长度, 且最大丢弃概率参数根据类平均队列长度调整, 保证延时区分。调度依据的队列权重值也是根据平均队列长度动态调整的, 从而更能反应实时网络状况, 保证带宽服务。中继路由器的包丢失率可以表示为包到达率和队列缓冲区的函数<sup>[9]</sup>, 所以通过缓冲区管理也可以保证丢失率区分。

### 2 DPAA 模型分析及形式化表示

PDS 模型定义其类间 QoS 约束如下:

$$\frac{\beta_i}{\beta_j} = \frac{\alpha_i}{\alpha_j}, \text{ 即 } \frac{\beta_i}{\alpha_i} = \frac{\beta_j}{\alpha_j} \quad 1 \leq i, j \leq N \quad (1)$$

其中 $\beta_i$ 表示类 $i$ 获得的服务质量,  $\alpha_i$ 表示类 $i$ 的服务区分参数。若 $\alpha_i$ 表示类 $i$ 的服务延时区分参数, 则有 $\alpha_1 > \alpha_2 > \dots > \alpha_i > \dots > \alpha_N > 0$ 。这个模型框架的控制目的就是保证映射到高级别类的数据流得到不低于低级别类数据流的服务质量, 且各类的服务质量按服务区分参数保持一定的比例关系。

根据 Little 公式, 在单队列系统中, 使用表 1 的符号定义, 有如下结论:

$$q_i = s_i d_i \quad (2)$$

式(2)没有关于队列中包到达或离开的任何假设, 所以在有多个逻辑类队列的系统中根据式(1)的约束有如下结论成立:

$$\frac{q_i(t)}{q_j(t)} = \frac{s_i(t)d_i}{s_j(t)d_j} \quad (3)$$

在按动态 WFQ 进行包调度的 PDS 模型中根据式(3)可以得到如下结论:

$$\frac{\omega_i(t) \delta_i}{\omega_j(t) \delta_j} = \frac{q_i(t)}{q_j(t)} \quad (4)$$

设 $\sum_i \omega_i(t) = 1$ , 若取 $\omega_i(t)$ 为参考值, 根据式(4)可以把 $\omega_j(t)$ 表示为 $\omega_i(t)$ 的函数, 从而可以推导出

$$\omega_i(t) = \frac{q_i(t)}{\sum_{j=1}^N \frac{\delta_i}{\delta_j} q_j(t)} \quad (5)$$

式(5)表示根据当前的缓冲区长度动态调整类调度权重, 从而达到实时的网络带宽占用控制。它的计算复杂度为 $O(N)$ ,  $N$ 为区分服务中的流分类级别个数。

中继节点(路由器)的数据丢失率可以表示为缓冲区大小的函数<sup>[9]</sup>, 式(6)是类 $i$ 的丢失率函数表示:

表 1 文中用到的符号定义

| 符号                                           | 意义                             |
|----------------------------------------------|--------------------------------|
| $i, j$                                       | 类编号索引, $1 \leq i, j \leq N$    |
| $\mu_i$                                      | 类 $i$ 的包服务时间                   |
| $\lambda_i$                                  | 类 $i$ 的包到达率(包/秒)               |
| $\delta_i$                                   | 类 $i$ 的延时比例参数                  |
| $p_i$                                        | 类 $i$ 的队列包丢失概率                 |
| $q_i, q_i(t)$                                | ( $t$ 时刻)类 $i$ 的队列长度           |
| $\omega_i(t)$                                | $t$ 时刻类 $i$ 的包调度权重             |
| $q_{i\_max\_th}, q_{i\_min\_th}, q_{i\_max}$ | 类 $i$ 的队列最大阈值, 最小阈值, 丢弃概率的最大取值 |
| $li(t)$                                      | $t$ 时刻类 $i$ 的包丢失率              |
| $s_i, s_i(t)$                                | ( $t$ 时刻)类 $i$ 的吞吐量(包/秒)       |
| $d_i$                                        | 类 $i$ 中包的平均延时                  |
| $C_i(t)$                                     | $t$ 时刻类 $i$ 获得的带宽              |
| $C$                                          | 总的链路带宽                         |

$$l_i(t) = \pi(0)^i \left( \frac{\lambda_i}{\mu_i} \right)^{q_{i\_max\_th}/2+1} \left( \frac{(1-p_i)\lambda_i}{\mu_i} \right)^{q_{i\_max\_th}/2} \quad (6)$$

$$\text{其中 } \pi(0)^i = \left[ 1 + \sum_{j=1}^{q_{i\_max\_th}/2+1} \left( \frac{\lambda_i}{\mu_i} \right)^j + \sum_{j=1}^{q_{i\_max\_th}/2} \left( \frac{(1-p_i)\lambda_i}{\mu_i} \right)^j \right]^{-1}.$$

所以 DPAA 可以表示为一个在特定约束条件(式(1)约束)下的求解问题, 即求

$$\left. \begin{aligned} \omega_i(t) &= f(q_1(t), \dots, q_i(t), \dots, q_N(t)) \\ l_i(t) &= g(p_{i\_max\_th}) \\ \overline{q_i(t)} &= h(p_{i\_max\_th}) \end{aligned} \right\} \quad (7)$$

式中 $p_{i\_max\_th}$ 的动态值及相应的 $\overline{q_i(t)}$ 值和 $\omega_i(t)$ 值, 其中 $\overline{q_i(t)}$ 是 $t$ 时刻类 $i$ 的平均队列长度,  $\omega_i(t)$ 的计算中取 $i$ 队列的 $q_i(t)$ 。

DPAA模型的控制结构如图 1 所示。通过一个列表记录各个类队列的平均长度, 当有数据包到来时, 根据式(6)计算各队列的包丢失率; 然后根据式(1)的约束, 确定从哪个队列丢弃包, 并调整各队列的 $p_{i\_max\_th}$ ; 根据式(5)计算各队列的调度权重, 其中的队列长度取此刻的平均队列长度。这样就同时考虑了各个区分服务质量因素。

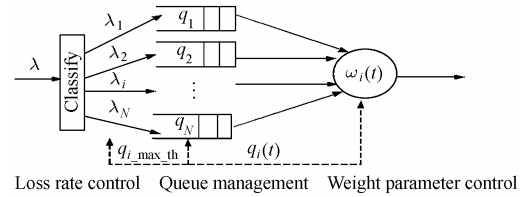


图 1 DPAA 的控制结构

Fig.1 Control structure of DPAA

### 3 DPAA 实现策略

一个自适应系统应该能够根据流量负载情况的变化而采取不同的调度管理策略, 我们根据流量负载状况把系统划分为 3 种模式: (1) 轻负载模式: 总流量负载小于系统通信能力。这种模式下除了信道导致的包丢失外, 其余包都会被系统缓冲区接纳; (2) 饱和模式: 总流量负载刚好等于系统通信能力。这种模式下所有进入系统的数据包都被接纳, 且可以通过调度和队列管理保证带宽和延时服务区分; (3) 重负载模式: 总流量负载超过了系统通信能力。这种模式下到达系统的数据包不能被缓冲区完全接纳, 存在一个丢弃策略选择问题。

在本文中, 我们主要考虑系统重负载模式下的 DPAA 实现策略, 用一个一维列表  $\text{Length}[i]$  记录实时测得的类 $i$ 的平均队列长度。

#### 3.1 队列管理及延时保证

初始时对系统缓冲区根据既定级别类进行分配, 采用类 RED 的方式进行队列管理。RED 机制有 3 个关键参数( $q_{max\_th}$ ,

$q_{\min\_th}$ ,  $\bar{q}_{avg}$ ), 文献[10]证明了这几个参数的最优取值范围分别为(0.6, 0.2, 0.4), 所以设定各队列的初始阈值为此值。然后随着流量负载的变化, 为了控制队列长度频繁变化抖动, 我们通过 $q_i(t)$ 对 $p_{i\_max}$ 进行实时调节, 从而使队列平均长度接近 $(q_{i\_max\_th} + q_{i\_min\_th})/2$ 。具体步骤如下:

(1) 取

$$p_{i\_max}(t + \Delta t) = p_{i\_max}(t) + \frac{2p_{i\_max}(t)}{q_{i\_max\_th} - q_{i\_min\_th}} \cdot \left[ q_i(t + \Delta t) - \frac{q_{i\_max\_th} + q_{i\_min\_th}}{2} \right] \quad (8)$$

这样使 $p_{i\_max}$ 随着队列长度 $q_i(t)$ 线性变化;

(2) 类 $i$ 的队列延迟为

$$d_i(t) = \frac{q_i(t)}{C_i(t)} = \frac{q_i(t)}{\omega_i C} = \frac{q_i(t)}{C \sum_{j=1}^N \frac{\delta_j}{C} q_j(t)} = \frac{\delta_i \sum_{j=1}^N q_j(t)}{C \sum_{j=1}^N \delta_j} \propto \delta_i$$

(因为 $\forall j \in [1, N]$ , 根据式(1)的约束有 $q_j(t)/\delta_j$ 近似相等, 所以有上面的比例结论成立)。

### 3.2 包丢弃机制

当 $\sum_i \lambda_i > B$ 时, 有新数据包到达时要进行策略性丢弃。根据式(6)可计算得到类 $i$ 此刻的丢失率, 在式(1)的约束下选择要丢弃包的队列, 调整 $q_{i\_max\_th}$ , 保证丢失率的服务区分比例。因为我们采用了类RED的队列管理机制, 所以队列中的包丢弃满足RED的特性: 即 $p_i$ 是 $p_{i\_max}$ 的线性函数。在3.1节中, 我们证明了所采取策略的类队列延时与 $\delta_i$ 成正比变化关系, 且 $p_{i\_max}$ 随 $q_i(t)$ 正比变化, 所以 $p_i$ 也随 $q_i(t)$ 正比变化, 且与其成线性关系。

### 3.3 调度机制

有数据包到来时用 $\text{Length}[i]$ 代替式(5)中的 $q_i(t)$ , 实时计算WFQ机制中类 $i$ 的权重因子 $\omega_i(t)$ 。因为根据实时平均队列长度计算调度权重值, 所以对有较高优先级的1帧流能够得到优先调度, 比较适用于视频流的传输质量保证。我们在下一节的仿真实验中通过与静态WRR的性能比较, 证明了这种自适应转发机制的有效性。

### 3.4 具体实现步骤

(1) 初始传输数据时, 节点路由器按照既定支持的服务类别分配缓冲区, 即按优先级静态分配带宽, 并设置各个队列的初始阈值为(0.2, 0.6);

(2) 当系统处于重负载模式时, 每到达一个数据包计算并记录各队列的平均长度 $\text{Length}[i]$ ;

(3) 按式(6)计算各类队列的丢失率, 若不满足式(1)则调整 $p_{i\_max\_th}$ ,  $p_{i\_min\_th}$ 取其1/2; 并根据式(8)动态调整 $p_{i\_max}$ , 其中的 $q_i(t)$ 取此时的 $\text{Length}[i]$ ;

(4) 根据式(5)计算包调度权重因子, 其中的 $q_i(t)$ 取此时

的 $\text{Length}[i]$ 。

### 3.5 算法复杂度分析

DPAA 机制的核心就是通过比例参数约束对队列阈值、调度权重因子进行实时调整, 即通过式(6), 式(8)和式(5)对每一个类的参量进行计算, 所以整个机制的复杂度是RED/WFQ 机制的 $N$ 倍,  $N$ 为系统支持的数据类别数(一般这个值小于12, 目前的路由器中通常设为4)。

## 4 实验仿真

我们通过实验仿真测试 DPAA 的性能, 实验环境选用了ns2.6, 网络拓扑结构如图2。节点ER1, CR3和ER2是区分服务区域, 其中CR3-ER2是瓶颈链路, DPAA 部署在CR3上。测试流选用UDP封装的应用流, 背景流是TCP流, 数据包大小为1000byte。

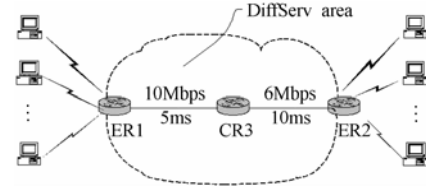


图2 仿真实验网络拓扑结构

Fig.2 Simulated network topology

我们基于3个类(类1, 类2和类3)进行了测试, 优先级依次是类1<类2<类3。设定其服务区分参数分别为 $\alpha_1=2$ ,  $\alpha_2=3$ ,  $\alpha_3=5$ , 且 $\delta_1=5$ ,  $\delta_2=3$ ,  $\delta_3=2$ 。下面分别从带宽、延时和丢失率三方面测试DPAA的性能。

**实验1** 流量负载为90%, 类1, 类2, 类3的流量分别为30%, 30%, 30%, 实验时间90s, 重复实验10次, 图3为DPAA与WTP/PLR( $\infty$ ), WRR/DROP TAIL(使用了ns2.6中的DiffServ实现)的实验结果显示。

**实验2** 流量负载为90%, 类1、类2、类3的流量分别为30%, 30%, 30%, 在[20s, 40s]时间段内改变流量负载比例: 增加类3的流量, 减少类1的流量, 实验时间90s, 重复实验10次, 图4为DPAA与WTP/PLR( $\infty$ ), WRR/DROP TAIL的实验结果显示。

通过对比两个实验的结果可以发现WRR, WTP对流量负载的变化比较敏感, 在实验2中[20s, 40s]增加类3的流量减少类1的流量, 即类负载比例变化时, WRR对类3的数据包进行了大量的丢弃, 丢弃率高于类1和类2, 但对吞吐量的服务比例保证没有受到干扰; WTP/PLR( $\infty$ )中类3的延时降低了, 类3和类2丢弃率下降了。DPAA由于基于网络状况的自适应调整, 优先保证丢失率, 所以丢失、延时要优于WRR, 在无突发流量时达到了与WTP/PLR( $\infty$ )相似的结果, 对突发流量不敏感, 但是对吞吐量的服务保证还不理想。

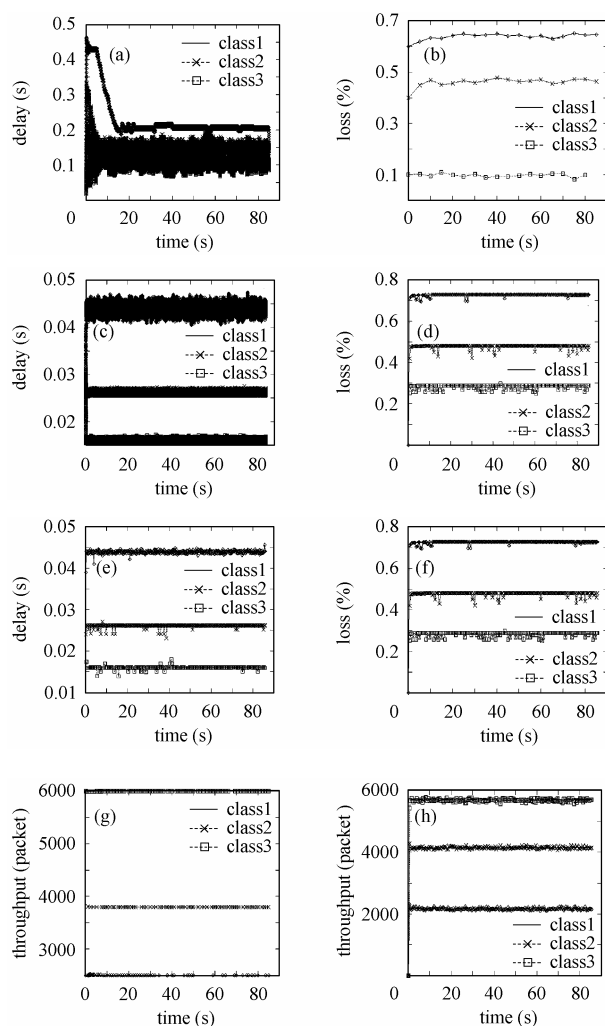


图3 实验1结果

(a),(b)是 WRR/DROP TAIL 的延时、丢失结果,  
(c),(d)是 WTP/PLR( $\infty$ )的延时、丢失结果,  
(e),(f)是 DPAA 的延时、丢失结果,  
(g)是其吞吐量表示, (h)是吞吐量表示

Fig.3 results of simulation 1

(a),(b) are the delay and loss rate of WRR/DROP TAIL,  
(c),(d) are the delay and loss rate of WTP/PLR( $\infty$ ),  
(e),(f) are the delay and loss rate of DPAA,  
(g) is its throughput, (h) is its throughput

## 5 结束语

本文基于 PDS 模型提出一种在 IP 网络系统重负载模式下同时考虑带宽、延时和丢失率约束支持的 DPAA 队列调度机制,通过仿真实验证明了这种机制的有效性。DPAA 机制是基于 RED 和 WFQ 的改进,所以与现存的各种缓冲区管理调度算法有较好的兼容性,实现容易,且对流量负载变化不敏感,适用于有突发性的实时视频流等的传输。但是由于 DPAA 机制在实时调度参数调整时先满足丢失约束,所以吞吐量服务保障方面不如 WRR,这是我们下一步研究工作的改进目标。

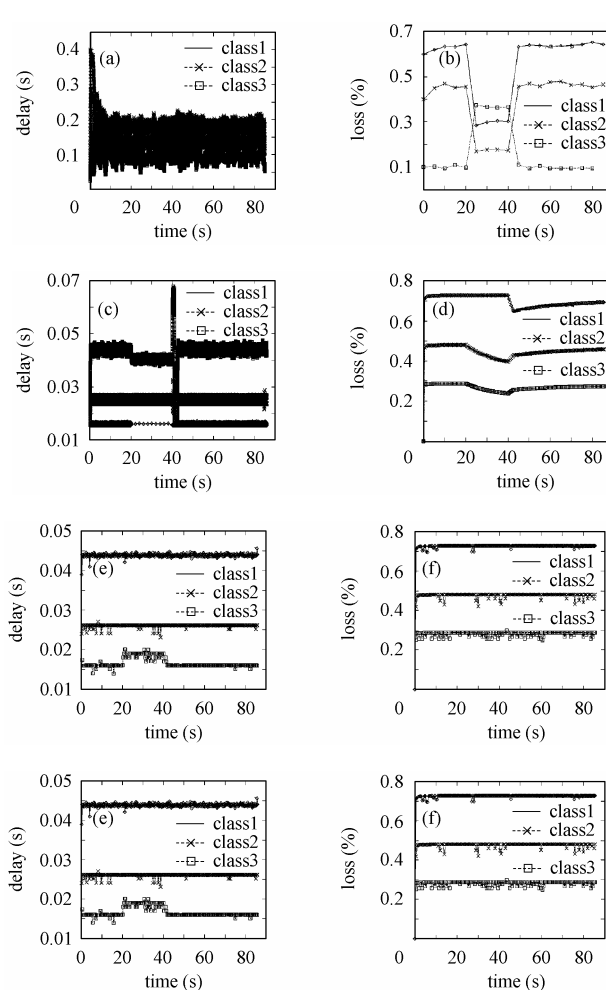


图4 实验2结果

(a),(b)是 WRR/DROP TAIL 的延时、丢失结果,  
(c),(d)是 WTP/PLR( $\infty$ )的延时、丢失结果,  
(e),(f)是 DPAA 的延时、丢失结果,  
(g)是吞吐量表示, (h)是吞吐量表示

Fig.4 Results of simulation 2

(a),(b)are the delay and loss rate of WRR/DROP TAIL,  
(c),(d)are the delay and loss rate of WTP/PLR( $\infty$ ),  
(e),(f)are the delay and loss rate of DPAA,  
(g) is its throughput, (h) is its throughput

## 参考文献

- [1] Blake S, Black D, Carlson M, Davies E. An architecture for differentiated services, IETF RFC 2475, December 1998.
- [2] Nichols K, Blake S, Baker F, Black D. Definition of the differentiated services field (DS Field) in the IPv4 and IPv6 headers, IETF RFC 2474, December 1998.
- [3] Braden R, Clark D, Shenker S. Integrated services in the internet architecture: an overview, IETF RFC 1633, 1994.
- [4] Bernet Y, Ford P, Yavatkar R, *et al.* A framework for integrated services operation over diffserv networks, IETF RFC 2998, November 2000.
- [5] Dovrolis C, Stiliadis D, Ramanathan P. Proportional

differentiated services: Delay differentiation and packet scheduling. *IEEE/ACM Trans. on Networking*, 2002, 10(1): 12–26.

[6] Dovrolis C, Ramanathan P. Proportional differentiated services, Part II: Loss rate differentiation and packet dropping. IEEE Eighth International Workshop on Quality of Service, Pittsburg PA, 5–7 June, 2000: 53–61.

[7] Samyukta Sankaran, Ahmed E Kamal. A combined delay and throughput proportional scheduling scheme for differentiated services. IEEE Global Telecommunications Conference, Taipei, Taiwan, 17–21 Nov., 2002, 2: 1910–1914.

[8] Li J S, Mao C S. Providing flow-based proportional differentiated services in class-based diffServ routers. *IEEE Proceedings on*

*Communications*, 2004, 151(1): 82–88.

[9] 向渝, 雷维礼, 黄顺吉. 区分服务模型中的一种动态队列管理算法. 通信学报, 2003, 24(10): 56–61.

[10] Floyd Sally. RED: Discussions of setting parameters. <http://www.aciri.org/floyd/REDparameters.txt>. 1997.

郭小清: 女, 1976年生, 博士生, 研究领域包括视频流传输、中继节点QoS调度算法、视频流量模型等.

吴介一: 男, 1941年生, 教授, 博士生导师, 研究领域包括模式识别、信息系统集成、计算机网络等.

张飒兵: 女, 1964年生, 高级工程师, 研究领域包括分布式计算、计算机网络等.