

## 一种新的分布式 QoS 控制动态资源调配算法

徐名海 糜正琨 彭玉旭

(南京邮电大学通信与信息工程学院 南京 210003)

**摘要** 该文提出了一种新的分布式环境下用于 QoS 控制的基于模糊逻辑的动态资源调配(Fuzzy logic based Dynamic Resource Allocation, FDRA)算法。分布式环境采用集中式带宽代理(Bandwidth Brokers, BB)与分布式 BB 相结合的控制模式。分布式 BB 基于滑动窗口和滑动指针机制,在分配资源不足时向集中式 BB 请求追加资源,在占用资源高于需求一定阈值时滞后释放部分追加资源。追加资源块和释放资源块的大小分别由集中式 BB 和分布式 BB 根据当前网络负荷状态,采用模糊逻辑确定。模糊逻辑算法引入新的基于数据源特性的隶属度函数生成方法,提高决策的有效性。仿真结果证明,该文提出的算法性能优于已有文献的算法。

**关键词** 分布式 QoS 控制, 动态资源调配, 模糊逻辑, 滞后释放释放

中图分类号: TN915.07

文献标识码: A

文章编号: 1009-5896(2006)09-1670-05

## A New Dynamic Resource Allocation Algorithm for Distributed QoS Control

Xu Ming-hai Mi Zheng-kun Peng Yu-xu

(Communication and Information Engineering College, Nanjing University of Post and Telecommunications, Nanjing 210003, China)

**Abstract** A new Fuzzy-logic based Dynamic Resource Allocation algorithm (FDRA) used for QoS control is proposed in this paper. In the hierarchically distributed architecture with multiple Bandwidth Brokers (BB), using sliding window and sliding pointer based mechanism, when the allocated resource is in short, a request is issued by the distributed BB to the centralized BB for additional resource. When the resource kept by a distributed BB is over the actual need to some extent, some allocated resource will be released with hysteresis. The size of additional allocated resource and released resource is determined by the centralized BB and the distributed BB respectively. The determination is concluded with a fuzzy logic algorithm taking into account of the current network load status. In order to improve the effectiveness of the decision, a new method of fuzzy mapping function generation is proposed, which is based on the statistical characteristics of the data sources. Simulation results show that the proposed algorithm is superior to the existing ones.

**Key words** Distributed QoS control, Dynamic resource allocation, Fuzzy logic, Resource release with hysteresis

### 1 引言

区分服务(DiffServ)是 IP 网络重要的 QoS 技术,然而它只能提供相对优先级的 QoS。为了保证确定的 QoS 性能要求,DiffServ 域必须增设网络带宽资源分配和网络流量控制功能,为此引入了网络资源管理器<sup>[1-5]</sup>的概念。在 IETF 标准中,称之为带宽代理(BB)<sup>[1, 2]</sup>,其功能是管理和控制 DiffServ 域的带宽资源,包括域内和域间的资源管理和流量控制。但是,至今为止 IETF 只考虑集中式 BB 结构,即每个域配置一个 BB,负责该域的资源分配和管理。这种集中控制模式很难适应大规模网络的需求,其主要问题是:(1)网络边缘接受的所有资源请求都需转交集中设置的 BB 处理,由此将显著增加网络负荷;(2)全域资源请求由单一 BB 处理,将形成处理瓶颈,难以满足会话型和交互型业务流的实时性要求。为此,人们提出了不同形式的分布式 BB 控制

模式<sup>[6-8]</sup>,其基本思想是由集中设置的 BB(cBB)维护全域的拓扑和资源信息,由位于边缘的众多的分布式 BB(eBB)维护与其相关的路径信息。对于边缘到达的资源请求,直接由 eBB 进行处理,只有当 eBB 资源不够时,才会向 cBB 请求追加资源,cBB 负责对全域资源进行动态优化分配控制。文献[8]给出了兼顾资源利用率和计算开销的量化分配资源的思路,即当 eBB 提出追加资源请求时,cBB 并不是根据请求的数量精确地追加分配,而是分配规定大小的一块资源给 eBB,以避免频繁的资源追加请求,资源使用完毕后立即释放。这样可有效地减少处理开销。但是,文献并没有很好地解决以下两个问题:

(1) cBB 量化分配的资源块应该是多大?文献只是简单地考虑资源请求的需要,没有考虑网络当前负荷状态,可能导致资源调配的不公平性。

(2) 追加分配的资源如何释放?文献采用简单的立即释放策略,没有考虑后续到达请求的趋势,可能导致资源追加频率和处理开销的增加。

本文针对上述两个问题提出了两个算法。一个是根据网络实时负荷状态进行动态资源块分配的模糊逻辑算法，一个是基于趋势概率分析的资源追加请求和滞后释放算法，合称为基于模糊逻辑的动态资源分配算法(Fuzzy-logic based Dynamic Resource Allocation, FDRA)。在模糊逻辑算法中，本文提出了一种新的模糊隶属度函数生成方法：二次高斯概率率。仿真结果表明，所提出的算法解决了分布式 BB 结构尚未解决的技术问题，具有良好的性能。本文第 2 节给出模型假设和定义，第 3 节提出算法及其描述，第 4 节为仿真结果，最后提出了进一步研究的方向。

## 2 模型假设与定义

分布式 BB 控制结构如图 1 所示。图中，资源代理(Resource Agent, RA)是位于边缘路由器中的资源控制实体，预先分配有一定数量的资源池，相当于 eBB 的角色，主要处理动态资源请求、设备控制和测量信息采集。当预分配的资源不够并且到达的请求优先级较高时，它将向上级动态资源控制器(Dynamic Resource Controller, DRC)提出资源追加请求。DRC 相当于 cBB 的角色，负责处理由 RA 提出的资源追加和释放请求。RA 和 DRC 位于控制平面。ER 和 CR 分别是传输域的边缘路由器与核心路由器，位于数据平面。

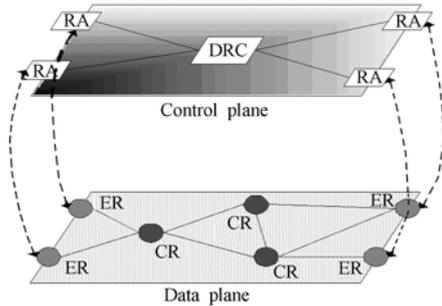


图 1 分布式控制结构

Fig. 1 Distributed controlling structure

假设数据平面采用 DiffServ over MPLS 技术，每个 RA 管理控制与对应 ER 相关的一组 LSP。为了描述 LSP 或链路的负荷状态，定义 LSP/链路负荷函数如下。

定义 1 设  $C$  为 LSP 或链路的带宽， $m, \sigma_{mes}^2$  分别是该 LSP 或链路上已有数据流量占用带宽的均值与方差，则 LSP 或链路负荷函数为

$$\eta = 1 - 2 \int_m^c \frac{1}{\sqrt{2\pi}\sigma_{mes}} \cdot \exp\left(-\frac{(x-m)^2}{2\sigma_{mes}^2}\right) dx \quad (1)$$

其中  $m = \frac{1}{N_s} \sum_{i=1}^{N_s} R_i$ ,  $\sigma_{mes}^2 = \frac{1}{(N_s-1)} \sum_{i=1}^{N_s} (R_i - m)^2$ 。

## 3 资源动态分配和释放算法

### 3.1 问题的描述

分布式环境下的动态资源控制所要解决的问题可以描述为：DRC 维持全域链路负荷状态信息库，对于来自 RA

的每次资源追加请求，DRC 根据网络资源使用状况确定追加的资源；在占用资源高于需求一定阈值时，RA 需滞后释放部分追加的资源，以使得开销尽可能小，资源利用率尽可能大。

对于资源追加分配问题，如果 DRC 分配给 RA 的资源块过小，则可能由于较高的资源请求到达率，RA 将频繁地向 DRC 请求追加资源，从而造成处理开销增大；反之，如果 DRC 分配给 RA 的资源块过大，则会由于当前 RA 占用过多资源，造成网络总体资源利用率的下降。由此可知，DRC 量化分配的资源块大小必须与网络当前负荷状况相适应。

对于资源释放问题，如果 RA 追加资源使用完毕后立即收回，则当下一次资源请求到达时，RA 很可能需要再次发出资源追加请求，从而使处理开销增加；如果追加分配给 RA 的资源不及时收回，显然将使其它 RA 的资源追加请求得不到满足，从而影响网络性能，降低了网络资源利用率。

### 3.2 资源追加分配算法

设  $R_{pvn}^p, R^p, R_{req}^p, R_{pol}^p$  分别是第  $p$  路径上预先分配的资源、已经使用的资源、当前请求的资源 and 额外分配的资源总和， $r$  是第  $p$  路径上当前到达或离去流的资源大小， $R_{hys}^p$  和  $R_{thres}^p$  分别是第  $p$  路径上的滑动指针和滑动窗口大小，remainc 是剩余共享资源。使用基于滑动窗口和滑动指针的机制可以与流到达或离去的动态性相适应，减少 eBB 和 cBB 之间的交互，提高 eBB 的工作效能。

如图 2 所示，在两种情形下，RA 将向 DRC 发出资源追加请求。首先，当一个流到达时，若  $R^p + r \geq R_{pvn}^p + R_{pol}^p$ ，RA 根据一定策略向 DRC 请求追加资源，DRC 将利用模糊逻辑根据网络当前负荷状况计算确定本次追加分配的资源块( $R_{req}^p$ )大小，并更新  $R_{pol}^p$  和 remainc。同时维持一个滑动指针  $R_{hys}^p$ ，其初值设为滑动窗口  $R_{thres}^p$  的一半，每到达一个请求，若其请求的资源大小为  $r$ ，向左滑动指针： $R_{hys}^p = R_{hys}^p - r$ 。

其次若  $R_{hys}^p \leq 0$ ，说明请求到达占主导，它包括两层含义：(1)到达的请求需要较多的资源，即单个请求的  $r$  较大；(2)到达的请求数量比较多。为了提高接纳率和资源利用率，由 RA 向 DRC 发出追加资源请求。具体处理流程如图 3 所示。

### 3.3 资源滞后释放算法

资源释放的基本思想是允许 RA 占有一部分剩余追加资源，只有当这部分资源在某个时间段一直没有使用，才根据网络当前负荷状态释放部分资源。这个时间段的长度决定于滑动窗口大小和流到达或离去的速率。



图 2 资源追加与滞后释放示意图

Fig. 2 Sketch of resource allocation and release with hysteresis

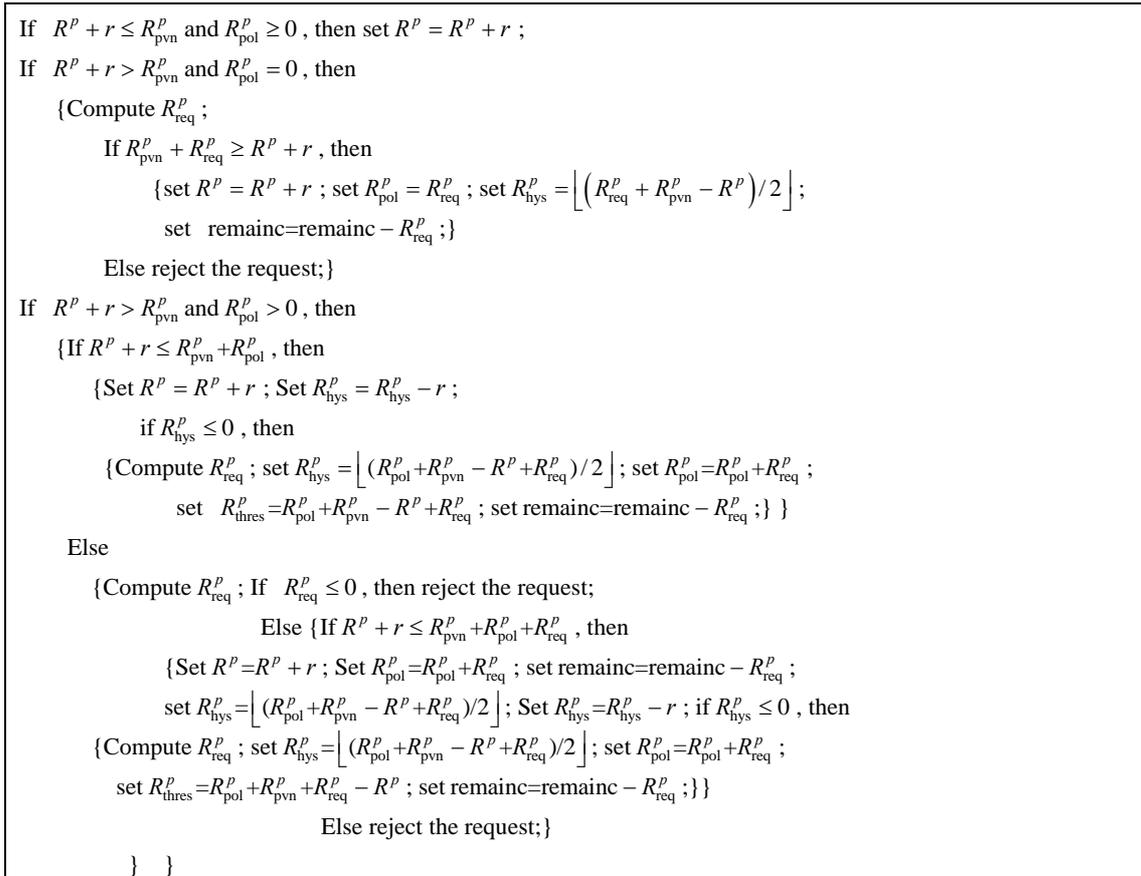


图 3 流到达的处理流程

Fig.3 Process progress when flow arrives

如图 2 所示,当某个流离去时,若其所占用的资源大小为  $r$ ,向右滑动指针:  $R_{hys}^p = R_{hys}^p + r$ ,当  $R_{hys}^p$  滑动超出右端即  $R_{pvn}^p + R_{pol}^p$  时,预示业务流的离去是主要趋势,应该释放资源,此时由 RA 根据网络负荷,利用模糊逻辑计算本次释放的资源  $R_{rel}^p$ ,并更新  $R_{pol}^p$  和 remainc。 $R_{rel}^p$  计算的基本思路是:在此 RA 控制的剩余资源范围内,若当前剩余共享资源较少,则多释放一些资源;若当前剩余共享资源较多,可以少释放一些资源。具体处理流程如图 4 所示。

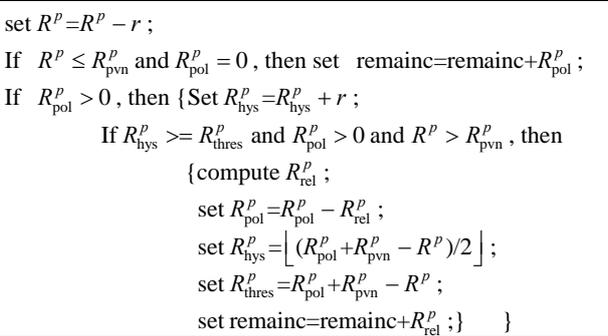


图 4 流离去时的处理流程

Fig.4 Process progress when flow departs

### 3.4 基于模糊逻辑的 $R_{req}^p$ 和 $R_{rel}^p$ 计算

3.4.1  $R_{req}^p$  的计算  $R_{req}^p$  计算的基本思路是,当网络负荷相对较大时,  $R_{req}^p$  取较小值;当网络负荷相对较小时,  $R_{req}^p$  取较大值;如果网络负荷居中,则  $R_{req}^p$  也取中等值。这里网络负荷用负荷函数描述。

基于模糊逻辑的  $R_{req}^p$  计算包括负荷函数  $\eta$  的模糊化、模糊推理和去模糊化 3 个步骤。

(1) 负荷函数  $\eta$  模糊化  $\eta$  模糊化的目的是将负荷函数测量值  $\eta$  转换成模糊化向量。算法考虑 3 个正规模糊集合:高(high)、中(medium)和低(low),并提出如下的二次高斯概率率法确定这 3 个模糊集对应的隶属度函数。

(a) 计算参数变量序列  $\eta$  的均值  $u$  和方差  $\sigma_\eta$ ;

(b) 利用均值  $u$  和方差  $\sigma_\eta$ ,得到一次高斯函数  $\frac{1}{\sqrt{2\pi}\sigma_\eta} \cdot \exp\left(-\frac{(x-u)^2}{\sigma_\eta^2}\right)$ ;

(c) 根据给定的取值概率  $p_1, p_2$ , 求解积分方程

$$\left. \begin{aligned} \int_0^{x_1} \exp\left(-\frac{(t-u)^2}{\sigma_\eta^2}\right) dt &= p_1 \\ \int_0^{x_2} \exp\left(-\frac{(t-u)^2}{\sigma_\eta^2}\right) dt &= p_2 \end{aligned} \right\} \quad (2)$$

得到  $x_1, x_2$ ;

(d) 令  $x_1, u, x_2$  分别为 3 个模糊集对应的主值基准点,记  $k = (\sqrt{3}/\sigma_\eta)^2 = 3/\sigma_\eta^2$ , 分别以  $u = x_1, u = u, u = x_2$  代入  $u_A(x) = \exp(-k(x-u)^2)$ , 得到对应 3 个模糊集的二次高斯函数;

(e) 修正,对于“低”模糊集,二次高斯函数左瓣在  $[0, x_1]$  区间内单调递增,即意味着参量值愈小,属于“低”模糊集的隶属度愈小,不符合客观实际情况;同理,对于“高”模糊集,二次高斯函数右瓣在  $[x_2, 1]$  区间内单调递减,也不符合客观

实际情况；因此将“低”模糊集高斯函数的左瓣和“高”模糊集高斯函数的右瓣修正为 1，由此得到 3 个模糊集的隶属度函数。

与常规确定的模糊集隶属度函数相比，上述二次高斯概率率充分考虑具体数据源的特性，据此模糊化的效果更逼近实际。

当  $\eta$  取值为 3 个主值基准点时，其模糊化所得的隶属度如表 1 所示。

表 1 主值基准点的模糊化隶属度  
Tab.1 Membership of benchmark points

	$x_1$	$\mu$	$x_2$
High(H)	$x_{31} = \exp(-k(x_1 - x_2)^2)$	$x_{32} = \exp(-k(\mu - x_2)^2)$	1
Medium (M)	$x_{21} = \exp(-k(x_1 - \mu)^2)$	1	$x_{23} = \exp(-k(x_2 - \mu)^2)$
Low(L)	1	$x_{12} = \exp(-k(\mu - x_1)^2)$	$x_{13} = \exp(-k(x_2 - x_1)^2)$

(2) 模糊推理 模糊推理是模糊控制的核心，它模拟人类模糊推理能力。推理过程是基于模糊逻辑中的蕴含关系及推量规则来进行的。模糊控制规则由一系列“IF-THEN”型的模糊条件句所构成。模糊控制规则如表 2 所示。

表 2 模糊逻辑规则  
Tab.2 Fuzzy logic rules

If (负荷函数)	Low(L)	Medium(M)	High(H)
Then( $R_{req}^p$ 大小)	大(RB)	中(RM)	小(RS)

进一步设定  $R_{req}^p$  的模糊集隶属度如表 3 所示，即可构造模糊关系矩阵  $R$  为

$$R_i = (x_{i1}, x_{i2}, x_{i3})^T \circ (\mu_{i1}, \mu_{i2}, \mu_{i3}, \mu_{i4}, \mu_{i5}), \quad i = 1, 2, 3, \quad (3)$$

$$R = \bigcup_{i=1}^3 R_i = (r_{ij})_{3 \times 5}$$

其中  $\circ$  是模糊乘运算。

再将负荷函数值  $\eta$  的模糊化向量  $\tilde{\eta}$  和模糊关系矩阵进行模糊乘运算，即得到模糊决策  $b$  为

$$b = \tilde{\eta} \circ R = (b_1, \dots, b_5) \quad (4)$$

(3) 去模糊化 去模糊化的作用是根据模糊决策  $b$  计算得到等效带宽的确定值  $R_{req}^p$ 。加权平均法是一种比较理想的去模糊化方法，其计算公式为

$$R_{req}^p = b \bar{R}_{req}^p / \sum_{i=1}^5 b_i \quad (5)$$

$$R_{req}^p = \lceil R_{req}^p \rceil$$

其中  $\lceil x \rceil$  表示对  $x$  上取整。

表 3 设定的  $\tilde{R}_{req}^p$  模糊集隶属度  
Tab.3 Predefined membership of  $\tilde{R}_{req}^p$

$\tilde{R}_{req}^p$	1/40 remainc	1/20 remainc	3/40 remainc	1/10 remainc	1/8 remainc
大(RB)	$\mu_{11}$	$\mu_{12}$	$\mu_{13}$	$\mu_{14}$	$\mu_{15}$
中(RM)	$\mu_{21}$	$\mu_{22}$	$\mu_{23}$	$\mu_{24}$	$\mu_{25}$
小(RS)	$\mu_{31}$	$\mu_{32}$	$\mu_{33}$	$\mu_{34}$	$\mu_{35}$

其中  $\lceil x \rceil$  表示对  $x$  上取整。

3.4.2  $R_{rel}^p$  的计算 追加资源释放包括两种情况：

(1) 当实际使用的资源小于预先分配的资源时，还有追加分配的资源，此时把部分追加分配的资源释放掉，即  $R^p \leq R_{pvn}^p$  且  $R_{pol}^p > 0$  时释放资源；

(2) 当滑动指针滑动到窗口最大值时，释放资源，即  $R_{hys}^p \geq R_{thres}^p$  时释放资源。

$R_{rel}^p$  的推理规则与  $R_{req}^p$  相反，即网络负荷较大时， $R_{rel}^p$  取较大值；当网络负荷较小时， $R_{rel}^p$  取较小值；当网络负荷中等时， $R_{rel}^p$  取中等值。

根据这种思路，只需要对表 3 中的  $R_{req}^p$  作相应调整，即可得到  $R_{rel}^p$  的模糊集隶属度。根据相似的计算可以得到  $R_{rel}^p$ 。

3.4.3 计算复杂性分析 在确定模糊化参数和模糊关系矩阵  $R$  后，如果不考虑其基于网络负荷状态的自适应调整，对于每次资源追加请求，需要执行的计算是：计算  $\eta$  的模糊化向量  $\tilde{\eta}$ ；计算模糊决策  $b$ ；去模糊化得到  $R_{req}^p$ 。为了得到  $\tilde{\eta}$ ，最多需要进行 3 次简单指数函数计算，最少需要 2 次指数函数计算和 1 次比较运算。由于  $R$  是  $3 \times 5$  阶矩阵，为了计算模糊决策  $b$ ，需要 5 次取小取大运算。而计算  $R_{req}^p$  则需要进行 5 次乘法、8 次加法、1 次除法和 1 次取整运算。整个计算需要 32 个变量的存储空间。对于每次资源释放请求，所需要的计算量及存储空间与此相当。当然，如果考虑模糊化参数和模糊关系矩阵  $R$  的自适应调整，计算量会较大增加（主要是模糊化参数的调整），但是这种自适应调整不需要针对每一次的资源追加或释放请求，可以设定触发门限，只有在网络负荷状态的变化突破了设定的门限才进行参数的自适应调整，这样可以在保持算法有效性的前提下，大大降低算法的复杂性。

#### 4 仿真分析

仿真分析的目的是比较本文算法与文献[8]提出的 PoQ 算法的性能。

PoQ 的基本思想是将链路资源划分为若干 Quota，每个 Quota 都是大小相等的一块链路资源，Quota 的大小通常远高于平均的资源请求大小。资源分配工作于两种模式：Normal 模式和 Critical 模式。在 Normal 模式下，对于 eBB 的每次资源请求，DRC 都分配一个固定大小的 Quota，用完之后再释放；在 Critical 模式下，由于剩余共享资源较少，DRC 按照请求多少分配多少的原则进行资源分配，用完之后立即释放。因此，按照 PoQ 方案，链路数据库更新次数就包括在 Normal 模式下 Quota 的分配与释放次数以及在 Critical 模式下资源追加请求和释放的次数。

仿真拓扑结构如图 5 所示。图中，入口路由器  $I_1 \sim I_K$  与核心路由器  $R_1$  相联， $R_1$  到出口路由器  $E_1$  的链路为瓶颈链路， $I_1 \sim I_K$  到  $R_1$  的链路容量设为无穷大。从入口路由器到达的流服从到达间隔均值为  $1/\lambda$  的指数分布，流在系统中的滞留时间服从均值为  $1/\mu$  的指数分布。在仿真中，设定  $1/\mu = 900s$ 。设每个流请求一个单位的带宽，瓶颈链路容量

为  $C = 5400$ ，入口路由器数目  $K = 3$ 。

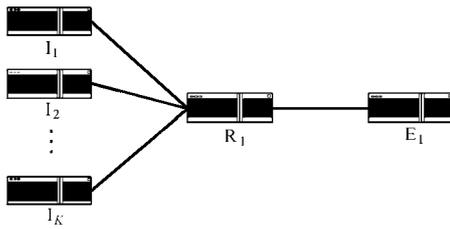


图 5 仿真拓扑结构

Fig. 5 Simulation topology

主要考虑两个指标：接纳率 RA 和链路数据库更新次数 NU。考虑到在初始状态系统的不稳定性，我们截取 6000 ~ 10000s 内的数据，设 AR 为到达流的总数，AC 为接纳的流的总数，对于 PoQ 方案而言，设  $G$  和  $L$  分别是分配 Quota 和收回 Quota 的次数， $M$  为在 Critical 模式下接纳的流的数目。对于本文所提出的方案(FDRA)，设 NReq 为请求追加资源的次数，NRel 为请求释放资源的次数。则有

$$RA = AC / AR$$

对于 PoQ， $NU = 2M + G + L$ 。

对于 FDRA， $NU = NReq + NRel$ 。

在仿真中，我们考虑 6 种平均负荷状态，其负荷函数值分别是 0.95, 0.96, 0.97, 0.98, 0.99, 1。首先计算采用 PoQ 算法时的上述两个指标，其中 Quota 分别取 30, 60, 90, 120, 150。然后计算采用本文算法时的上述两个指标，其中  $R_{pvn}^p = 1600$ ,  $p = 1, 2, 3$ 。图 6 给出了两种算法在 6 种负荷状态下的接纳率变化曲线。可以看到，两种算法的曲线变化趋势是一致的，即当负荷较低时，拒绝率很低，随着负荷的增加，接纳率呈下降趋势，但是无论 Quota 取多大的值，本文提出的 FDRA 算法的接纳率始终高于 PoQ 算法。

图 7 给出了 6 种负荷状态下瓶颈链路数据库更新次数的变化关系。由图可见，两种算法的趋势也是相同的，即当网络负荷上升时更新次数呈上升趋势，但是 FDRA 算法的数据库更新次数始终小于 PoQ 算法。

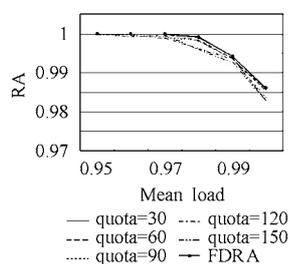


图 6 六种负荷状态下的接纳率曲线

Fig. 6 Accept ratio under six load level

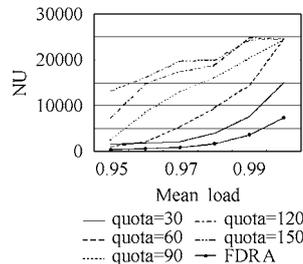


图 7 六种负荷状态下的链路数据库更新次数

Fig. 7 Updates of link database under six load level

从图 6 和图 7 可以看出，相对于 PoQ 算法而言，FDRA 算法有效地提高了接纳率，同时使由于瓶颈链路数据库更新带来的处理开销处于较低水平。这主要是因为，在 FDRA 算法中，系统能根据网络负荷状态请求或释放资源，当网络

负荷较大时，少请求一些额外的资源可以避免资源浪费，多释放一些资源可以保证接纳更多的流；当网络负荷较小时，多请求一些资源可以降低访问瓶颈链路的次数，少释放一些资源可以避免由于流的动态性带来的处理开销。

相对于 PoQ 方案而言，FDRA 有两大特点：使用滑动窗口和滑动指针来适配流的动态到达和离去；根据当前的负荷状态使用基于模糊逻辑的算法确定追加和释放的资源块大小。这样，FDRA 将对资源的控制与流到达的随机性以及网络实际负荷有机联系起来，克服了 PoQ 方案对资源控制在 Normal 模式下相对静态在 Critical 模式下频繁振荡的缺陷。所以 FDRA 相对于 PoQ 不仅提高了接纳率，还有效地降低了处理开销，保持了系统的稳定性。

## 5 结束语

本文研究了在分布式 QoS 控制环境下如何进行动态资源分配的问题，提出了根据网络负荷状态进行基于模糊逻辑的资源分配与资源滞后释放算法。仿真表明，该算法优于此前文献提出的分布式资源控制算法。尚待进一步研究的问题包括： $R_{req}^p$  和  $R_{rel}^p$  模糊隶属度取值的设定对算法性能的影响；如何根据流量工程确定  $R_{pvn}^p$  的初值；更复杂的网络条件下的算法性能分析。

## 参考文献

- [1] Reichmeyer F, Ong L, Terzis A, Zhang Lixia. A two-tier resource management model for differentiated services networks. INTERNET-DRAFT, November 1998.
- [2] Nichols K, Jacobson V, Zhang L. A two-bit differentiated services architecture for the Internet. RFC 2638, July 1999.
- [3] Khalil I, Braun T. Implementation of a bandwidth broker for dynamic end-to-end capacity reservation over multiple diffServ domains. MMNS 2001, 2001, LNCS 2216: 160-174.
- [4] ITU-T COM 13-D351. A carrier-class QoS solution framework for IP-based backbone networks. Geneva, 21 July-1 August, 2003.
- [5] ITU-T COM 13-D350. A carrier-class QoS solution framework for IP-based access network. Geneva, 21 July-1 August, 2003.
- [6] Giordano S, Salsano S, Van den Berghe S, et al.. Advanced QoS provisioning in IP networks: The European premium IP projects. *IEEE Communications Magazine*, 2003, 41(1): 30-36.
- [7] Engel T, Granzer H, Koch B F, et al.. AQUILA: Adaptive resource control for QoS using an IP-based layered architecture. *IEEE Communications Magazine*, 2003, 41(1): 46-53.
- [8] Zhang Zhi-Li, Duan Zhen-hai, Hou Y T. On scalable design of bandwidth brokers. *IEICE Transactions on Communications*, 2001, E84-B(8): 2011-2025.

徐名海：男，1977 年生，博士生，研究方向为 IP 网络 QoS。

糜正琨：男，1946 年生，教授，博士生导师，主要研究方向为 IP 通信网理论和技术。

彭玉旭：男，1977 年生，博士生，研究方向为流量建模与 Ad hoc 网络的路由。