

# 基于 DiffServ 模型的调度算法

鲍 慧, 赵生岗, 黄 霞

(华北电力大学电子与通信工程系, 保定 071003)

**摘要:** 区分服务(DiffServ)模型中不同队列调度算法对网络性能有不同的影响。该文介绍了 DiffServ 实现模型, 分析比较了目前 4 种典型队列调度算法的基本原理及性能特点。基于 OPNET Modeler 构建了采用不同调度算法的 DiffServ 仿真实现模型, 通过对各仿真结果的研究, 进一步验证了几种算法各自的优劣特性。

**关键词:** 区分服务模型; 服务质量; 调度算法; 网络仿真

## Queuing Algorithm Based on DiffServ Model

BAO Hui, ZHAO Sheng-gang, HUANG Xia

(Dept. of Electronic and Communication Engineering, North China Electric Power University, Baoding 071003)

**【Abstract】** In DiffServ QoS framework, queuing algorithm is the key factor that impacts the network performance. This paper introduces the DiffServ realization model, analyzes the basic principle of four main queuing algorithms and compares their characteristics, constructs the DiffServ simulation realization model with different queuing algorithms, validates and compares the different characteristics of queuing algorithms through the simulation.

**【Key words】** DiffServ model; Quality of Service(QoS); queuing algorithm; network simulation

### 1 概述

随着网络技术及多媒体技术的飞速发展, 越来越多的多媒体应用对网络提出了更高的要求, 语音、视频会议、远程控制等实时多媒体应用需要网络提供相应的服务质量(Quality of Service, QoS)保证。QoS是指网络针对某一类服务所能提供和达到的性能与质量, 具体可以量化为带宽、延迟、抖动、丢失率、吞吐量等性能指标<sup>[1]</sup>。

目前的 IP 网络对所有的数据包都是公平的, 只能提供“尽力而为”型服务, 这不能满足不同业务对服务质量的不同要求。为此, IETF 提出了 2 种不同的 IP QoS 体系结构: 综合服务模型(IntServ)和区分服务模型(DiffServ)。其中, DiffServ 具有良好的可扩展性, 适合于在大型主干网上提供 QoS 服务。队列调度机制是保证 DiffServ 实现的核心机制之一, 是解决多个业务竞争共享资源问题的有效手段。路由器中调度算法的主要作用是决定从等待队列中选择哪个分组进行发送, 主要影响带宽分配、延迟和延迟抖动等性能参数, 是实现网络服务质量的核心技术之一。

### 2 DiffServ 实现模型

DiffServ模型<sup>[2]</sup>是一个相对简单粗粒度的QoS控制模型, 其针对的是流聚合后的每一类QoS控制。DiffServ模型在边界节点将进入网络的单流分类、整形、聚合为不同的流聚集, 在IPv4头中的TOS域(IPv6头中的TC域)中为每个入流标识不同优先级的区分服务码点值DSCP(DiffServ Code Point)。核心节点则根据DSCP选择相应的逐跳行为PHB(Per Hop Behavior), 通过队列调度机制来完成对不同分组不同PHB的区分, 实现对不同用户、应用的服务区分。目前, IETF已经定义了4类PHB, 分别为尽力而为型(Best Effort, BE)、确保转发型(Assured Forwarding, AF)PHB、加速转发型(Expedited Forwarding, EF)PHB以及兼容IP优先级的类选择型(Class

Selector, CS)PHB。其中, AF PHB又包括了AF1, AF2, AF3和AF4 4个等级。

DiffServ 实现模型由边缘路由器和核心路由器的若干功能模块组成, 主要包括: 包分类模块, 用户流量检测(策略)模块, 标记模块, 缓冲区管理模块, 队列调度模块和流量整形模块。支持 DiffServ 的边缘路由器基本上要求支持上面的全部模块, 而中间节点可以不提供用户流量检测(策略)模块、标记模块。

### 3 调度算法分析

DiffServ路由器进行数据转发时, 需要通过队列调度算法决定转发哪个分组。目前支持DiffServ的主流调度算法有如下几种: 先进先出队列(FIFO)算法, 优先级队列(PQ)算法, 基于GPS的调度算法(FQ, WFQ), 基于轮询的调度算法(RR, WRR, DWRR), 基于WFQ改进的CBWFQ和LLQ算法等。下面对几种典型的算法进行分析<sup>[3-4]</sup>。

#### 3.1 先进先出队列(FIFO)调度算法

FIFO是最基本的队列调度算法, 它按照数据包到达路由器的先后次序进行传输。FIFO的最大优点在于实现简单, 缺点是不区分流、没有恶意流阻止功能等。

#### 3.2 优先级队列(PQ)调度算法

优先级队列调度算法提供了一个相对简单的支持区分服务类的方法。PQ把数据包分类放置到具有不同优先级的队列中, 再依次调度各优先级队列: 最先调度优先级最高的队列中的数据包, 当它为空时才调度次高级别的。依次类推, 直到比某个队列更高优先级的队列中没有数据包时, 才调度这个队列。每个队列以FIFO方式对包进行调度。PQ能保证高

**作者简介:** 鲍 慧(1962 - ), 女, 副教授, 主研方向: 宽带网及交换技术, 下一代网络; 赵生岗、黄 霞, 硕士研究生

**收稿日期:** 2007-11-24 **E-mail:** tzhao123@sohu.com

优先级的业务，但是会出现低优先级得不到服务的情况，即出现带宽“饥饿”现象。

### 3.3 加权公平队列(WFQ)调度算法

WFQ 调度算法根据 DSCP 来区分不同业务，根据每个队列不同的权重值调度每个队列，优先处理权重值高的队列，按照队列的权重值分配带宽。WFQ 是把 GPS 模型应用于网络系统的一种近似算法。GPS 模型是一个理想化的流模型，假定每一个队列中的业务元可以无限小，队列之间具有相同的优先级，因而调度器可以根据各队列的共享比例同时服务所有的队列，能使各业务流真正公平地共享服务器。但 GPS 模型在实际网络系统中是无法实现的，因为业务元的最小粒度是分组，而且该分组的服务不能被抢断。

WFQ 算法的基本思想是维护一个系统虚拟时间  $V(t)$ ，并为每一个队列维护一个虚拟开始时间标签  $S_i(t)$  (队列  $i$  头部分组的虚拟开始发送时间)和一个虚拟完成时间标签  $F_i(t)$  (队列  $i$  头部分组的虚拟完成发送时间)，当队列中有分组发送完毕，或者空队列有分组到达时分别计算如下式：

$$S_i(t) = \begin{cases} \max(V(t), F_i(t)) & \text{如果队列 } i \text{ 在前一时刻为空} \\ F_i(t) & \text{如果分组 } p_i^{k-1} \text{ 被发送完毕} \end{cases}$$

$$F_i(t) = S_i(t) + L_i^k / r_i$$

其中， $F_i(t)$  是队列  $i$  更新之前的虚拟完成时间； $L_i^k$  是当前队列  $i$  第  $k$  个分组  $P_i^k$  的长度； $r_i$  是预先定义的服务速率。当每次需要调度时，系统根据时间标签的大小选择一个分组。通过给用户链接提供最小带宽保证，进而保证了时延范围。

WFQ 克服了传统方法的缺点，可对单个业务流进行精细的控制，为每个业务流提供较低可计算的端到端时延保障，同时它能以公平的方式使得各业务流共享剩余的带宽资源。WFQ 的缺点是流量粒度过细，计算复杂度较高。

### 3.4 差值加权轮询队列(DWRR)调度算法

为解决 WRR 不能支持长度不同的包而出现的带宽分配不公平以及计算复杂的缺点，提出了 DWRR 调度算法。在 DWRR 算法中，为每个队列设置一个权值  $W$  和一个差值计数器  $DC$  (每次允许调度器访问队列的字节总数)。具体算法过程如下：(1)调度器访问每个非空队列，如果队列头部的包长度大于  $DC$ ，则调度器移动到下一个队列。(2)如果队列头部的包小于或等于  $DC$ ，则变量  $DC$  减少包长字节数，并传送包到输出端口。调度器继续输出包和减少  $DC$  值，直到队列头部的包长度大于变量  $DC$  值。剩余的  $DC$  值将作为信用值累加到下次轮询时使用。(3)如果队列输出包直到队列为空，则设置  $DC$  为零，此时调度器将服务下一个非空队列。

DWRR 结合了 WFQ 和 WRR 的优点，能够在输出端口按队列的权值为流提供相应的带宽分配，从而避免低优先级队列出现带宽“饥饿”现象。DWRR 能够支持长度变化的流，特别是突发流，并且该算法实现比 WFQ 简单。

CBWFQ 和 LLQ 算法与 FQ, WRR 一样都存在算法复杂、处理速度慢的缺点。

## 4 DiffServ 仿真实现

本文通过 OPNET Modeler 10.0<sup>[5]</sup> 构建了仿真模型，实现了 DiffServ 服务模型，并对目前几种主要的调度算法进行了仿真，根据仿真结果对算法性能特点进行比较分析。

### 4.1 网络拓扑

本次仿真采用了单路径网络拓扑结构，如图 1 所示。为了更直接地分析仿真结果，本次仿真把 4 种业务全配置成为

视频业务。左边 4 台为 Video 客户端，经过交换机 A，边界路由器 A，边界路由器 B，再经过交换机 B 连接到最右边的 4 台 Video 服务器上。客户端与交换机以及交换机与路由器之间都是 10 Mb/s 带宽链路，路由器 A 与路由器 B 之间是 2 Mb/s 链路，此处设置为网络可能的瓶颈。设置网络瓶颈是为了当不同业务在边界路由器发生资源竞争时，更好地仿真出不同调度算法对不同等级业务的服务优劣。

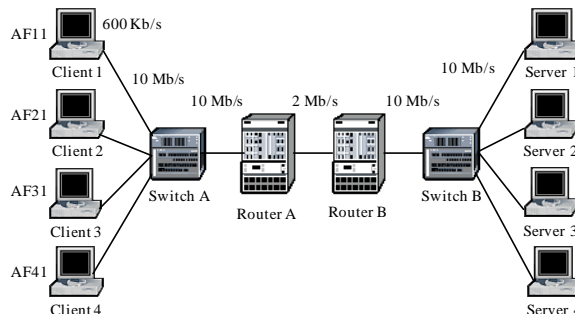


图 1 DiffServ 仿真实现模型

客户端与服务器 Video 业务都是连续的，速率为 600 Kb/s (30 frames/s, 2 500 B/frame)，这样当 4 个终端同时发送业务时将会在边界路由器出口处对带宽进行竞争。客户端 1~客户端 4 的优先级分别为 AF 级别的 AF11, AF21, AF31, AF41，优先级由低到高逐渐增加。

### 4.2 仿真结果

本次仿真共建立了 4 个不同的场景：FIFO, PQ, WFQ, DWRR，以分析网络采用不同调度算法对网络性能及业务的影响。为便于分析算法，不同场景中各队列缓冲区均为 1 MB。仿真结果搜集了网络的部分全局变量和节点的统计量，具体包括数据包延迟、数据包接收速率和丢包速率。

#### 4.2.1 网络仿真结果

对网络的全局统计变量有数据包的端到端延迟、数据包的接收速率和丢包速率，表征网络整体的性能特点，仿真结果如表 1 所示。

表 1 不同调度算法中网络各参数平均值

统计量	平均延迟/s	接收速率/(packet·s <sup>-1</sup> )	丢包速率/(packet·s <sup>-1</sup> )
FIFO	2.474 1	147.781 3	88.010 4
PQ	0.028 0	178.447 9	83.718 8
WFQ	1.863 0	151.807 3	84.666 7
DWRR	1.961 5	151.697 9	84.656 3

#### 4.2.2 各客户端仿真结果

各客户端仿真结果包括各客户端与其服务器之间数据包端到端延迟、数据包接收速率，表征不同调度算法对各客户端业务的影响。各客户端数据包端到端延迟如表 2 所示。

表 2 不同调度算法中各客户端包平均延迟 s

客户端	FIFO	PQ	WFQ	DWRR
Client 1	2.479 0	×	17.160 8	16.955 5
Client 2	2.476 5	0.036 7	0.215 1	0.209 5
Client 3	2.472 7	0.027 1	0.029 2	0.030 0
Client 4	2.478 6	0.018 1	0.025 3	0.024 6

### 4.3 仿真结果分析

由于网络瓶颈问题，数据包在路由器 A 中排队，因为路由器中设置了缓冲区大小限制，当缓冲区达到最大容量时，数据包将被丢弃，因此从图 2~图 5 所示的数据接收速率可以看出，各种算法场景中不同程度地出现了丢包现象。

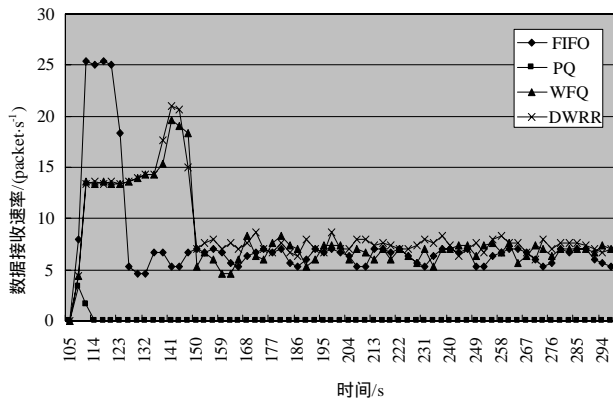


图2 客户端1的数据接收速率

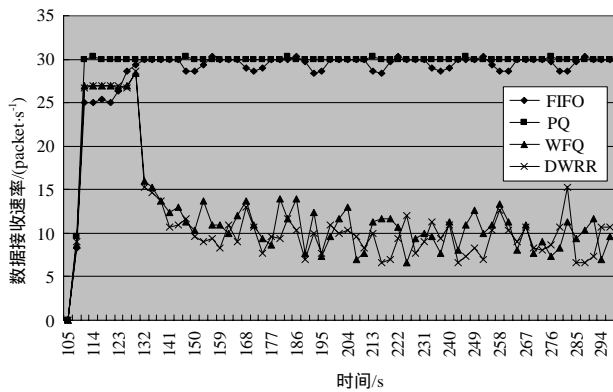


图3 客户端2的数据接收速率

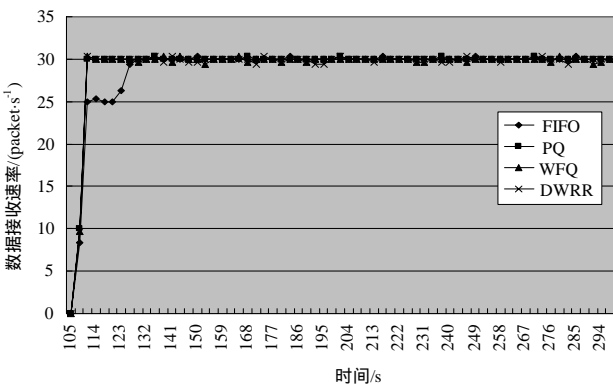


图4 客户端3的数据接收速率

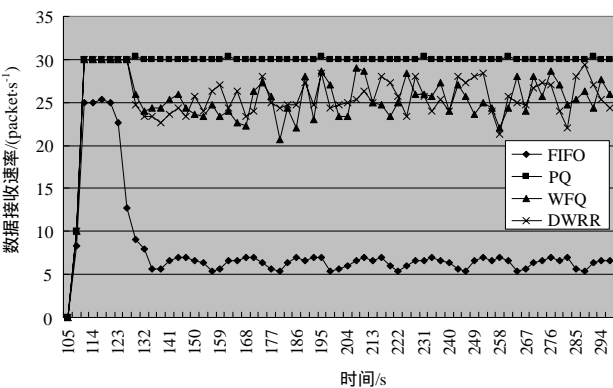


图5 客户端4的数据接收速率

由图2~图5可见：

(1)在采用FIFO的模型中，如图2~图5所示，在110s~120s之间所有客户端都维持在25 packet/s，而后Client2和Client3迅速增加到30 packet/s左右，而Client1和Client4迅

速降到5 packet/s~7 packet/s左右，从表1也可看出，FIFO模型出现明显丢包现象。从表2中可以看出，业务延迟保持在一个稳定的时间值，约为2.48s。这个时间值也就是一个数据包的最大延迟，是当这个包到达缓冲区且队列刚好要满时它前面所有数据包被处理的时间，也就是整个缓冲区队列被处理的时间。

(2)在采用PQ的模型中，虽然网络总接收速率较高，如表1所示，但从表2可以看出，优先级高的业务包延迟很小，而优先级别低的包延迟比较高。Client4的延迟只有18.1ms，Client3为27.1ms，Client2为36ms，3个高优先级客户端的数据包接收速率在110s以后都保持在30 packet/s左右，但业务1从第114s开始则几乎没有收到数据包，出现带宽“饥饿”现象，这说明了PQ的不公平性。

(3)在采用WFQ的模型中，如表2所示，Client4和Client3延迟比较小，分别为25.3ms和29.2ms，Client2延迟平均值约为215.1ms，虽然比PQ时延迟增大，但从表1和图2~图5中可以看出，WFQ模型中网络平均延迟最小，此时Client1的包接收情况得到了明显的改善，约在150s以后趋于平稳，略大于FIFO时的包接收量，由于有一定的权重值，分到了一定的带宽，而不像PQ时处于“饥饿”状态。

(4)在采用DWRR的模型中，DWRR调度算法与采用WFQ的模型性能很相近，但Client3中的延迟比WFQ略高，约为0.8ms；Client4、Client2和Client1比WFQ略低，分别为0.7ms、5.6ms和205.3ms。网络丢包速率比WFQ略低。

从以上的仿真结果分析得出：FIFO算法实现简单，处理速度快，但对每个数据包都同等对待，不能阻隔恶意流，不能对业务提供不同QoS保证。PQ算法可以为高优先级业务提供绝对的带宽保证，但是优先级别低的数据包根本没有机会分享到带宽。WFQ与DWRR算法性能比较接近，按权重不同公平分配带宽，不会出现优先级的独占带宽的情况，但算法的复杂度较高，配置相对困难，处理速度比较慢。

## 5 结束语

本文介绍了DiffServ模型的实现模型，重点研究了实现模型中的队列调度算法，并通过OPNET Modeler构建了DiffServ实现模型，对几种基本的调度算法进行了对比仿真研究。从仿真结果可以看出，在目前主要的几种算法中，还没有一种较为完善的算法，包括一些改进的算法，都不能同时满足不同业务的不同要求。本文的工作为进一步的算法分析和优化提供了有力依据。下一步的工作需要结合实现机制中的缓冲区管理等算法对调度算法做综合的仿真研究，以得到网络的全局最优实现模型，对现有算法作进一步的优化，使其能更好地满足不同网络条件下对不同业务需求的支持。

### 参考文献

- [1] 林 闯. 计算机网络的服务质量(QoS)[M]. 北京: 清华大学出版社, 2004.
- [2] Blake S, Black D, Carlson M, et al. An Architecture for Differentiated Services[S]. RFC 2475, 1998-12.
- [3] Sayenko A, Hamalainen T, Joutsensalo J, et al. Comparison and Analysis of the Revenue-based Adaptive Queuing Models[J]. Computer Networks, 2006, 50(6): 1040-1058.
- [4] 李晓花, 李小军. IP网络QoS几种排队算法的对比分析[J]. 四川理工学院学报: 自然科学版, 2006, 19(3): 81-84.
- [5] OPNET Online Documentation 10.0[Z]. OPNET Technologies Inc., 2003.