区分服务中 AF 类的一种调度算法 [1]

刘金梅 王思明

(暨南大学电子工程系 广州 510632)

摘 要: 该文根据区分服务中确保转发 (Assured Forwarding, AF) 类的特点, 设计了一种新的调度算法——公平加权轮循 (Fair Weighted Round Robin, FWRR) 算法. FWRR 是一种基于轮循、工作保持型、适于变长分组的调度算法. 它的实现简单, 算法复杂度为 $O(1)$. 仿真实验和数学分析表明, FWRR 算法不仅能够提供保证最小带宽的服务, 而且能够按比例分配剩余带宽, 适合用来调度区分服务中的 AF 类.

关键词: 区分服务, 确保转发, 公平加权轮循, 调度算法

中图分类号: TN919.3 **文献标识码**: A **文章编号**: 1009-5896(2003)12-1676-06

A Scheduling Algorithm for Assured Forwarding Aggregated Flows of Differentiated Services

Liu Jin-mei Wang Si-ming

(*Dept. of Electronic Engineering, Jinan University, Guangzhou* 510632, *China*)

Abstract According to the characteristics of Assured Forwarding (AF) aggregated flows of Differentiated Services(DiffServ), a new scheduling algorithm, named Fair Weighted Round Robin(FWRR) is proposed. FWRR is a work-conserving round robin scheduling algorithm for variable size packets. It has a very low implementation complexity of $O(1)$. Simulation results and mathematic analyses show that FWRR can make each queue at least occupy a minimum reserved bandwidth and share some extra bandwidth proportionally. Therefore, FWRR is a suitable scheduling algorithm for the AF flows of DiffServ.

Key words Differentiated Services(DiffServ), Assured Forwarding(AF), Fair Weighted Round Robin(FWRR), Scheduling algorithms

1 引言

作为一种在因特网上实现 QoS(Quality of Service) 保证的技术方案, 自 90 年代末以来, 区分服务 (Differentiated Services, 以下简称 DiffServ) 在国内外的网络研究领域中一直受到广泛关注. IETF(Internet Engineering Task Force) 已针对 DiffServ 发布了一些 RFC 文档 [1−9]. 目前已提出的 DiffServ 聚流有 3 种, 即 EF(Expedited Forwarding) 类, AF(Assured Forwarding) 组和 BE(Best Effort) 类. EF 类享有低时延、低抖动、低丢失率、保证带宽的服务, 这种 "三低一保证" 的服务类似于虚拟专线服务, 是目前所定义的最高级别的 DiffServ 类型. AF 组可以享有确保带宽的服务. 根据所提供带宽比例的不同, AF 组又分为 4 个不同的 AF 类. AF 类只要求保证带宽和丢失率, 不涉及延迟和抖动, 其服务原则是: 无论网络是否发生了拥塞, 用户都能至少得到所约定的最低限量的带宽, 并且, 网络若有空余资源, 用户可以得到一定的额外带宽. BE 类只能享有缺省的尽力而为型服务. 通常, EF 类、AF 组和 BE 类之间的调度可以采用优先级调度 [2], WFQ(Weighted Fair Queucing) [2,10], WF^2Q(Worst-case Fair weighted

[1] 2002-07-22 收到, 2002-12-05 改回
"211 工程" 重点学科建设项目 "计算机信息与通信技术" 资助课题

Fair Queueing)[2,10] 等调度算法. AF 组中各个 AF 类之间的调度也可以使用一些现有的调度算法, 如 WFQ, WF^{2}Q 等. 这些调度算法大多具有很好的时延性和公平性, 但实现起来复杂, 而 AF 类对时延和抖动的要求不高, 所以, 我们根据 AF 类的特点, 设计了一种新的调度算法 —— FWRR(Fair Weighted Round Robin) 算法. 它不仅易于实现, 而且能够为 DiffServ 中各 AF 类提供确保带宽的服务, 能够按一定比例分配剩余带宽, 满足带宽分配公平性的要求.

本文的后续部分是这样安排的: 第 2 节介绍了 FWRR 调度算法; 第 3 节对 FWRR 算法进行了编程仿真; 第 4 节是对 FWRR 的数学分析; 最后是结论部分.

2 公平加权轮循调度算法 —— FWRR 算法

2.1 FWRR 算法特点

FWRR 算法是一种基于轮循的、工作保持型、适用于变长分组的调度算法. 为什么将 FWRR 算法设计成是基于轮循的调度算法呢? 这是因为虽然基于轮循的调度算法的时延特性稍差, 但易于实现, 运算复杂度为 $O(1)$, 而且 DiffServ 并不需要为 AF 类的时延特性提供保证. 所以, FWRR 算法是基于轮循的. 在 FWRR 算法中, 轮循周期的长度是根据队列中等待分组的情况在一定范围内可变的, 所以, 它是工作保持型的调度算法. 另外, 由于在 DiffServ 中, 组成各 AF 类的分组来自多个具有不同的源 / 目的地址的微流, 这些微流的分组长度不尽相同. 所以, FWRR 算法的设计是基于变长分组的.

2.2 符号说明

在 FWRR 算法的描述中, 使用了下列符号 (其中, 分组长度和服务量均以字节为单位): N 为系统中的队列总数; W_i 为队列 i 的权值, 且 $\min(W_1, W_2, \cdots, W_N) = 1.0$; w_i 为队列 i 的瞬时权值; h 为当前服务队列中, 首部分组的长度; q_d 为当前调度片段内, 已得到约定服务量的队列个数; T 为 q_d 由 0 变为 N 的时间间隔, 称为调度片段; b_i 为本调度片段中, 队列 i 是否已得到约定的服务量, b_i 为 "1" 表示已达到本调度片段内的约定服务量, 要继续得到服务必须等待下一个调度片段; f_{w_i} 为队列 i 的连续服务量; $w_{i,k}$ 为某个 T 内, 队列 i 在接受了 k 次服务后的瞬时权值; n_i 为某个 T 内, 队列 i 已得到的服务次数; $\mathrm{S}_{i,k}$ 为队列 i 在当前 T 内, 在第 k 次服务中实际得到的服务量, 其中, $k = 1, 2, \cdots, n_i$; S_i 为某个 T 内, 队列 i 实际得到的总服务量; R_i 为一个 T 内, 队列 i 按权值应得到的总服务量; DI_i 为队列 i 在某个 T 内的服务偏差量; $L_{\min}$ 为系统中的最小分组长度; $L_{\max}$ 为系统中的最大分组长度; $L_{i,\max}$ 为队列 i 的最大分组长度; r 为总链路速率; r_i 为队列 i 的约定服务速率.

2.3 FWRR 算法描述

在 FWRR 调度算法中, 系统中的每个队列均具有一个固定权值 W 和瞬时参数 w. 调度器按照一定的服务规则轮流对各个队列进行服务, 从而使各个队列按照预定的权值比例共享出口链路带宽. 具体调度规则如下:

(1) 在一个新的调度片段开始时, 所有队列的瞬时参数均增加一个固定权值, 即

$$w_i = w_i + W_i, \quad i = 1, \cdots, N \tag{1}$$

(2) 每当调度器开始对队列 i 进行服务时, f_{w_i} 清零;

(3) 若队列 i 的首部分组得到了服务, 则进行下列运算: $w_i = w_i - h/L_{\max}$, $f_{w_i} = f_{w_i} + h$;

(4) 若 $f_{w_i} \le L_{\max} - 0.5h$, 则调度器继续对队列 i 进行服务, 否则, 调度器将结束本次轮循对队列 i 的服务, 转向服务下一个队列;

(5) 队列 i 变为空或 $w_i \le 0$ 时,

$$b_i = 1, \quad q_d = q_d + 1 \tag{2}$$

(6) 当 $q_d \geq N$ 时，当前调度片段结束，下一个调度片段开始.

3 数学分析

下面从数学角度来对 FWRR 算法的公平性加以讨论 (仍然采用第 2.2 节中定义的符号)。由 FWRR 调度算法，不难得出：

$$w_{i,n_i} = W_i - \sum_{k=1}^{n_i} s_{i,k}/L_{\max} \tag{3}$$

由 $S_{i,k}/L_{\max} \leq 1$，得 $w_{i,n_i} \geq W_i - n_i$。因为 $w_{i,n_i} \leq 0$，所以，$n_i \geq W_i$，即在一个调度片段 T 内，队列得到服务的总次数不会低于它的权值.

3.1 服务公平指数 (SFI, Service Fairness Index)[14]

设在 T 内的某一时刻，队列 i 得到了 $k(k \leq n_i)$ 次服务，队列 j 得到了 $l(l \leq n_j)$ 次服务。根据 FWRR 算法，可知：若 $r_i \neq r_j$，不失一般性，设 $r_i > r_j$，且由于 $W_i/W_j = r_i/r_j$，所以 $W_i > W_j$.

(1) 当 $\sum_{m=1}^{k} s_{i,m}/r_i \geq \sum_{m=1}^{l} s_{j,m}/r_j$ 时，由 $|\sum_{m=1}^{k} s_{i,m} - \sum_{m=1}^{l} s_{j,m}| \leq 1.5L_{\max}$，得

$$\sum_{m=1}^{k} s_{i,m} \leq \sum_{m=1}^{l} s_{j,m} + 1.5L_{\max} \tag{4}$$

故

$$\sum_{m=1}^{k} s_{i,m}/r_i - \sum_{m=1}^{l} s_{j,m}/r_j \leq (1/r_i - 1/r_j) \sum_{m=1}^{l} s_{j,m} + 1.5L_{\max}/r_i \tag{5}$$

由 $r_i > r_j$，得

$$\mathrm{SFI} = \left| \sum_{m=1}^{k} s_{i,m}/r_i - \sum_{m=1}^{l} s_{j,m}/r_j \right| < 1.5L_{\max}/r_i \tag{6}$$

(2) 当 $\sum_{m=1}^{k} s_{i,m}/r_i < \sum_{m=1}^{l} s_{j,m}/r_j$ 时，由 $|\sum_{m=1}^{l} s_{j,m} - \sum_{m=1}^{k} s_{i,m}| \leq 1.5L_{\max}$，得

$$\sum_{m=1}^{l} s_{j,m}/r_j - \sum_{m=1}^{k} s_{i,m}/r_i \leq (1/r_j - 1/r_i) \sum_{m=1}^{k} s_{i,m} + 1.5L_{\max}/r_j \tag{7}$$

所以

$$\mathrm{SFI} = \left| \sum_{m=1}^{k} s_{i,m}/r_i - \sum_{m=1}^{l} s_{j,m}/r_j \right| \leq (1 - r_j/r_i) \sum_{m=1}^{k} s_{i,m}/r_j + 1.5L_{\max}/r_j \tag{8}$$

当 r_i 与 r_j 十分接近时，$(1 - r_j/r_i) \sum_{m=1}^{k} s_{i,m}/r_j$ 的值很小，特别地，当 $r_i = r_j$ 时，有

$$\mathrm{SFI} = \left| \sum_{m=1}^{k} s_{i,m}/r_i - \sum_{m=1}^{l} s_{j,m}/r_j \right| \leq 1.5L_{\max}/r_i \tag{9}$$

3.2 DI_i

DI_i 是我们为研究方便而新提出的一个参数，它可以从一定程度上衡量 FWRR 算法提供确保带宽服务的能力．在 FWRR 算法中，虽然 T 的值并不是固定不变的，但不管 T 值的大小如何，都有 $0 \le |R_i - S_i| < L_{i,\max}$，由 $DI_i = |R_i - S_i|$，得

$$0 \le DI_i < L_{i,\max} \tag{10}$$

DI_i 的最大值不超过队列的最大分组长度，保证了 FWRR 算法公平分配带宽的能力．另外，$DI_i \ge 0$，说明 FWRR 算法可以提供具有最小保证带宽的服务．

综上所述，FWRR 算法是一种适合用于 DiffServ 中各 AF 类的调度算法．

4 仿真实验

仿真工具采用 ns-2 系统 [11,12]，按照第 2 节所述的 FWRR 算法进行编程仿真．仿真实验采用如图 1 所示的网络结构．

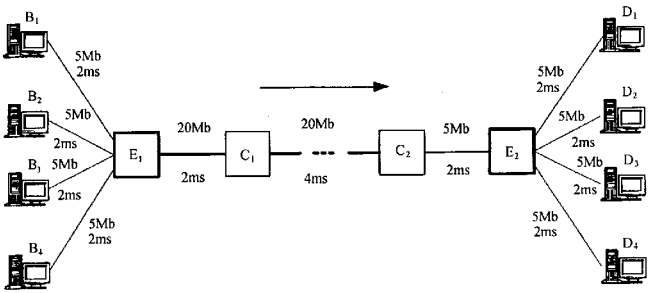

图 1　仿真实验网络结构图

图中，B_1, B_2, B_3 和 B_4 均为源端主机，D_1, D_2, D_3 和 D_4 为对应的目的端主机，E_1, E_2 为 DiffServ 边缘节点，C_1, C_2 为 DiffServ 内部节点．由 B_i 到 D_i 的信息流在边缘节点 E_1 处被标记为 $AF_i (i = 1, 2, 3, 4)$，它们分别组成 4 个独立的物理队列 AF_1，AF_2，AF_3 和 AF_4，队列权值分别为 4.0, 3.0, 2.0 和 1.0．各节点均采用 FWRR 调度算法，各队列均分配有足够大的缓冲空间 (大于最大队长)．

4.1 仿真实验 1

本仿真实验中，在仿真的起始时刻，B_1, B_2, B_3 和 B_4 均以 4Mb/s 的速度生成 On/Off 时间均为 10ms、形状参数 α 为 1.5 的 Pareto 流 (这里，由于 DiffServ 聚流服从自相似分布 [13]，所以使用自相似流来模拟 AF 流)，其分组长度分别为 1000, 1200, 1500 和 300byte．

仿真时间为 105s．得到如图 2 所示的带宽曲线．

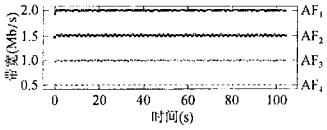

图 2 AF_i 在 $C_2 \sim E_2$ 链路中的实际带宽 (仿真实验 1)

各信息流的实际带宽分别大约为 2Mb/s, 1.5Mb/s, 1Mb/s, 0.5Mb/s，可见，它们是按照预先分配的权值 (4.0, 3.0, 2.0, 1.0) 来共享输出链路带宽的，与原始带宽无关. 这说明，FWRR 算法可以实现对“贪婪”流的约束，使有效带宽按约定的权值来分配. 只要保证链路带宽不低于某个阈值，FWRR 算法就可以为每个信息流提供一个最小保证带宽，这一点与 DiffServ 中 AF 类的服务要求是一致的.

4.2 仿真实验 2

本实验是用来测试 FWRR 算法能否按照权值比例分配剩余带宽. 仍然采用如图 1 所示的网络结构. B_1, B_2 和 B_3 在仿真起始时刻即开始发送分组，B_4 在仿真运行到 30s 时才开始发送分组. 其它仿真条件与实验 1 相同.

各信息流实际占用带宽情况如图 3 所示.

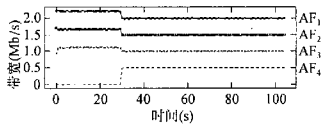

图 3 AF_i 在 $C_2 \sim E_2$ 链路中的实际带宽 (仿真实验 2)

由图 3 可以看出：

(1) 在 $t = 0 \sim 30$s 时，AF_4 的实际带宽为零，AF_1, AF_2 和 AF_3 的实际带宽分别约为 2.2Mb/s, 1.7Mb/s, 1.1Mb/s . 这说明，5Mb/s 的有效带宽在 $AF_1 \sim AF_3$ 之间按权值 (分别为 4.0, 3.0, 2.0) 比例进行了分配;

(2) 自 $t = 30$s 之后，经过一个短暂的过渡时间后，$AF_1 \sim AF_4$ 的实际带宽分别变为 2Mb/s, 1.5Mb/s, 1Mb/s, 0.5Mb/s，有效带宽在 4 个信息流之间被重新按比例进行了分配.

以上仿真实验说明，FWRR 算法不但能提供确保最小带宽的服务，而且可以实现剩余带宽的按比例分配. 所以，FWRR 算法可以作为 DiffServ 中各 AF 类之间的调度算法.

5 结 论

本文提出了一种新的调度 DiffServ 中各 AF 类的 FWRR 算法. 与其它能够用来调度 AF 类的算法 (如 WFQ 、 WF^2Q) 相比，它的实现更为简单，运算复杂度仅为 $O(1)$. 仿真实验和数学分析表明，FWRR 算法具有较好的公平性，能够提供具有最小保证带宽的服务，并能够按一定比例分配剩余带宽，满足 RFC2475 和 RFC2597 中规定的 DiffServ AF 类的服务要求. 此外，作为一种基于轮循、工作保持型、适用于变长分组的调度算法，FWRR 算法虽然不能提供十分严格的时延和抖动保证，但仍然可以用于其它像分组长度不固定、对时延和抖动要求不高、只对吞吐量有严格要求的情况.

参 考 文 献

[1] Blake S, Black D, Carlson M, Davies E, Wang Z, Weiss W. An architecture for differentiated services. RFC2475, December 1998.

[2] Heinanen J, Baker F, Weiss W, Wroclawski J. Assured forwarding PHB group. RFC2597, IETF, June 1999.

[3] Jacobson V, Nichols K, Poduri K. An expedited forwarding PHB. RFC2598, IETF, June 1999.

[4] Nichols K, Blake S, Baker F, Black D. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC2474, IETF, December 1998.

[5] Brim S, Carpenter B, Faucheur F L. Per hop behavior identification codes. RFC2836, IETF, May 2000.

[6] Huston G. Next steps for the IP QoS architecture. RFC2990, IETF, November 2000.

[7] Davie B, Charny A, Bennet J C R, Benson K, Le Boudec J Y, Courtney W, Davari S, Firoiu V, Stiliadis D. An expedited forwarding PHB (Per-Hop Behavior). RFC3246, IETF, March 2002.

[8] Charny A, Bennet J, Benson K, Boudec J, Chiu A, Courtney W, Davari S, Firoiu V, Kalmanek C, Ramakrishnan D K. Supplemental information for the new definition of the EF PHB (Expedited Forwarding Per-Hop Behavior). RFC3247, IETF, March 2002.

[9] Armitage G, Carpenter B, Casati A, Crowcroft J, Halpern J, Kumar B, Schnizlein J. A delay bound alternative revision of RFC 2598. RFC3248, IETF, March 2002.

[10] Bennett J C R, Zhang H. WF^2Q: Worst-case fair weighted fair queueing. Proceedings of IEEE INFOCOM, San Francisco, U.S., 1996: 120–128.

[11] [美]Welch B B 著，王道义，乔陶鹏译校. Tcl/Tk 组合教程. 北京：电子工业出版社，2001 年 1 月第 1 版，2–95.

[12] ns-2, http://www.isi.edu/nsnam/ns.

[13] [美]Stallings W 著，齐望东，薛卫娟，傅麒麟，胡谷雨译，谢希仁校. 高速网络— TCP/IP 和 ATM 的设计原理. 北京：电子工业出版社，1999 年 12 月第 1 版，154–173.

[14] Stiliadis D, Varma A. A general methodology for designing efficient traffic scheduling and shaping algorithms. Proceedings of IEEE INFOCOMM, Kobe, Japan, 1997: 326–335.

刘金梅：　女，1975 年生，教师，硕士，研究方向为计算机网络信息系统及应用.

王思明：　男，1941 年生，教授，研究方向为计算机网络信息系统及应用、电子信息技术模仿社会等.