

文章编号: 1001-0920(2006)09-1045-05

基于加权公平的DiffServ服务质量模型的研究

夏利¹, 杨宏², 张鹏¹, 王光兴¹

(1. 东北大学 教育部暨辽宁省流程工业综合自动化重点实验室, 信息科学与工程学院, 沈阳 110004; 2. 天津理工大学 计算机科学与工程系, 天津 300191)

摘要: 针对在核心节点实现的主动队列管理(AQM)可以预见和防止拥塞, 并且能够较公平地分配带宽, 介绍了一种加权公平的主动队列管理算法, 提出一个基于公平性的拥塞控制机制, 即将这种AQM算法部署在可提供QoS服务的区分服务模型中, 最后介绍仿真过程并分析该模型的性能

关键词: 区分服务; SCALE-WFS; 公平性

中图分类号: TP393 **文献标识码:** A

On Weighted Fairness-based DiffServ QoS Model

XIA Li¹, YANG Hong², ZHANG Peng¹, WANG Guang-xing¹

(1. Key Laboratory of Process Industry Automation of Ministry of Education, College of Information Science and Engineering, Northeastern University, Shenyang 110004, China; 2. School of Computer Science and Engineering, Tianjin University of Technology, Tianjin 300191, China Correspondent: XIA Li, Email: xial@mail.neu.edu.cn)

Abstract: The problem of active queue management (AQM) implemented in core nodes is discussed to predict and prevent congestion and allocate bandwidth in a fair manner. A weighted fairness guaranteed AQM algorithm is introduced, and then a fairness-based congestion control mechanism is presented in which the AQM approach is deployed in DiffServ model providing QoS guarantees. The simulations and analysis on the performance of this model are described.

Key words: DiffServ; SCALE-WFS; Fairness

1 引言

随着网络技术的快速发展和网络规模的不断扩大, 基于尽力而为(best-effort)的Internet已不能满足多媒体应用和其他非传统应用对网络传输质量的要求。DiffServ是IETF提出的QoS模型, 由于其简单有效、易扩展等特性, 已成为未来Internet主干网的核心技术之一^[1], 并得到越来越广泛的应用。

拥塞控制是实现QoS保证的措施之一, 而目前互联网上采用的拥塞控制方法基本上均为基于端点TCP层的拥塞窗口控制方法。这种方法对于UDP流不起作用, 且对不同往返时间(RTT)和不同带宽要求的TCP流之间也不公平, 进而导致DiffServ网

络中带宽分配的不公平性问题^[2,3]。

利用基于路由器的主动队列管理算法(AQM)进行拥塞控制并且保证网络资源分配的公平性成为当前计算机网络中研究的热点问题^[4-8]。SCALE-WFS算法是一种加权公平的分布式主动队列管理算法, 主要解决各种业务流共享AF(Assured Forward)时的公平性问题, 不仅能提供加权公平保证服务, 而且其主要功能集中在边缘节点的结构与DiffServ非常匹配^[3]。因此, 本文提出一种基于加权公平性的拥塞控制机制, 可将扩展加权公平排队算法应用到区分服务模型中以改善该模型的公平性。该机制无需改动DiffServ整体结构, 利用其原有的

收稿日期: 2005-07-11; 修回日期: 2005-09-12

基金项目: 国家863计划项目(708-4-5); 辽宁省自然科学基金项目(20042012)。

作者简介: 夏利(1962—), 女, 四川乐山人, 副教授, 博士生, 从事网络QoS理论及路由算法的研究; 王光兴(1937—), 男, 哈尔滨人, 教授, 博士生导师, 从事第三代移动通信、ATM网络管理等研究。

分类器、测量器、整形器等功能,只需修改其队列管理和队列调度模块,便可实现各个业务流之间共享带宽的加权公平性

2 相关技术

2.1 DiffServ 模型

DiffServ 模型主要包括边缘节点的分类器、流量调节器和核心节点的逐跳行为 (PHB) 转发。边缘节点根据用户与 ISP 之间的资源预留信息和服务等级约定参数 (如流轮廓速率) 将进入网络的单流分类、整形、聚合为不同的聚集流,聚集流的信息存储在每个 IP 包头的 D S 字段,称为 DSCP (differentiated services codepoint); 内部节点在转发 IP 包时根据包头的 DSCP 值提供特定服务质量的调度转发,称为 PHB。

为实现内部对聚集流提供特定服务质量的调度转发,队列管理对 DiffServ 而言至关重要。在路由器中采用一种 AQM 算法,即在缓存溢出之前就按一定策略丢弃分组,进而实现在保证较高吞吐量的基础上减小端到端延迟。

2.2 加权公平主动队列管理算法

加权公平队列管理算法 (SCALE-WFS)^[3] 是一种不需保存业务每个流状态而能提供各种流之间加权公平带宽共享的主动队列管理机制,其结构如图 1 所示,包括入边缘路由器、核心路由器和出边缘路由器。

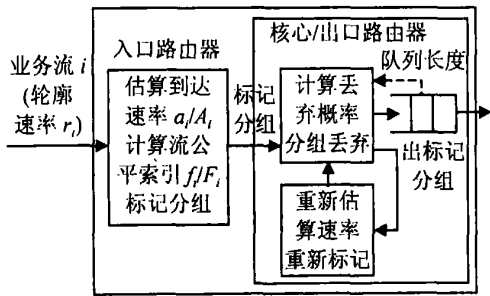


图 1 SCALE-WFS 结构

首先,在进入 DiffServ 域之前,标记器根据轮廓速率 r_i 向每个数据流打上标记,该标记表示用户与

ISP 之间或两个 ISP 之间协商的特定的端到端吞吐量期望值。然后,路由器对数据流中的分组进行概率转发,并对每个分组根据流到达速率 a_i 计算出流公平索引 f_i ,并进行再标记,用于在拥塞情况下决定分组的丢弃概率 P_i 。

路由器的每条输出链路保留一个 FIFO 队列,到达的分组或者入队或者丢弃。如果它们的标记 $(f_s, \dots, f_i, \dots, f_n)$ 近似相同,则其丢弃概率也相同,这样就可以用一个汇聚流公平索引 F_j 来表示,而不管每个数据流的 D 号。即数据流可以按照标记分成 $m + 1$ 个汇聚组 $(AG_0, \dots, AG_i, \dots, AG_m)$,并且编入由路由器维护的汇聚表中,因此这个表不会太大,不会占用过多的存储器资源。每个汇聚流 AG_j 丢弃概率的计算均基于 F_j ,汇聚到达速率 A_j 和链路公平索引 L 。

路由器在丢弃之后需要根据原标记 f_i/F_j 和丢弃概率 P_i/P_j 对每个入队的分组重新标记流公平索引。

3 基于加权公平性的 DiffServ 模型

DiffServ 中采用的是典型的 AQM 算法——RED 算法,不能保证公平性。下面提出一种机制,通过在 DiffServ 模型中采用一种加权公平主动队列管理算法,从而在保证网络可扩展性的同时,又保证网络的服务质量,特别是能够满足公平性需求。

把加权公平主动队列管理算法部署在区分服务模型中,如图 2 所示。在该模型中,可以借助 DiffServ 中原有的功能完成业务流汇聚、打标记和测量等功能。即在区分服务网络的入口路由器中,根据自身测量器将到达的数据包打上 DSCP 标记的同时打上速率标记。将头部含有速率标记和 DSCP 的数据包传送到核心路由器,核心路由器根据包头中 DSCP 值进行调度转发,同时根据速率标记查找对应 AG 表项,计算链路公平索引 L 和丢弃概率 P ,实现数据包入队和丢弃。这样,即可在不改变区分服务网络结构的基础上完成加权公平主动队列管理算法与区分服务网络模型的结合。

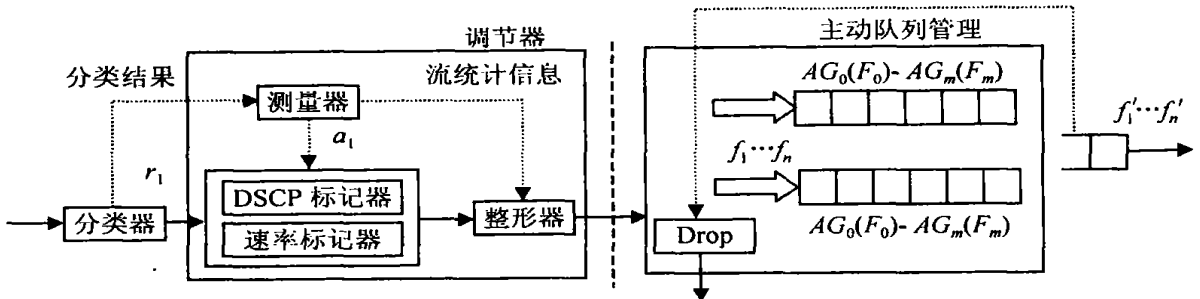


图 2 基于公平性的 DiffServ 模型

3.1 公平索引

在该模型中, 定义了 3 个公平索引: 流公平索引 f_i , 汇聚组公平索引 $AG(F_j)$, 输出链路公平索引 L . 假设数据流 i 的轮廓速率为 r_i , 到达速率为 a_i , 输出链路的容量为 C , 则流公平索引为

$$f_i = a_i / r_i \quad (1)$$

在边缘路由器可以通过时间滑动窗口或指数平均等方法来估算 a_i, f_i 表示当前流速率距离期望的吞吐量的差别

对于输出链路, L 的计算基于如下 max-min 公平等式:

$$r_i \times \min(L, f_i) = C, \text{ if } \sum_{i=1}^n a_i > C, \quad (2)$$

$$L = \max_{i=1, \dots, n} f_i, \text{ if } \sum_{i=1}^n a_i \leq C. \quad (3)$$

其中 $i = 1, \dots, n$ 表示 n 个流经过该链路, 为简便起见, 每条链路只有一个输出缓冲器. 在边缘路由器根据 n, r_i, f_i 解方程求出 L . L 表示当前链路的负载和作为共享该链路的流之间的公平性描述

对于 L , 流 i 的丢弃概率和新入队的分组公平索引分别为

$$p_i = \max(0, 1 - \frac{L}{f_i}),$$

$$f_i = f_i(1 - p_i), \forall i \in [1, \dots, n] \quad (4)$$

其中 f_i 表示 $\min(f_i, L)$, 它是流 i 的出队速率与轮廓速率的比. 如果满足式(4), 则认为获得 max-min 公平. 每个分组在从当前核心 / 出口路由器出队之前用 f_i 重新标记

假设所有输入流分为 $m + 1$ 个汇聚组 $AG_0 \sim AG_m$, 每个汇聚组中的所有数据流都具有相近的公平索引 f_i , 即汇聚组 AG_j 由数据流 f_i 组成, $i \in [s, t], f_i \cong F_j$. 则可以通过下列方程求解 L :

$$R_j \times \min(L, F_j) = C,$$

$$\text{if } \sum_{j=0}^m A_j > C, \quad (5)$$

$$L = \max_{j=0, \dots, m} F_j, \text{ if } \sum_{j=0}^m A_j \leq C. \quad (6)$$

其中 $R_j = \sum_{i=s}^t r_i$ 和 $A_j = \sum_{i=s}^t a_i$ 分别为第 j 个汇聚组 $AG_j = \{f_s, \dots, f_i, \dots, f_t\}$ 的轮廓速率和估算的到达速率, 通过 A_j / F_j 可计算出 R_j . L 通过迭代法求出, 不同的是用汇聚组代替了数据流, 计算 L 的复杂度减低了. 通常, 可变的汇聚数 $m + 1$ 均比数据流数小的多. 所以, 如果核心路由器在汇聚级上精确计算 L 将会容易得多. 基于 L , 可计算出 AG_j 的丢弃概率

和新标记为

$$P_j = \max(0, 1 - \frac{L}{F_j}),$$

$$F_j = F_j(1 - P_j), \forall j \in [0, m] \quad (7)$$

每个属于 AG_j 的分组在从当前核心 / 出口路由器出队之前也用 F_j 重新标记

3.2 标记器

标记器存在于 DS 域的入口 / 出口边缘路由器. DiffServ 中标记器的作用不变, 本文只研究业务流中的两个丢弃优先级别的转发过程. 每一个业务流中的数据包根据轮廓速率标记为 N 或 OUT, 那么单个业务流 / 聚合业务流的索引可以分解为 f_{i_N} / F_{j_N} 和 f_{i_OUT} / F_{j_OUT} .

3.3 入口边缘路由器

在入口边缘路由器, 当有 n 个业务流到达路由器的输入端时, 链路总容量为 C , 对于该链路, 路由器为其维持一个业务流表, 并执行对到达数据包进行分类操作和估算业务流 i 中 N / OUT 数据包的速率 A_{i_N} 和 A_{i_OUT} 的任务. 该模型中 N 和 OUT 的聚合标记与 DiffServ 的标记方法一样, 即超过轮廓速率的数据包打上 N 标记, 反之打上 OUT 标记. 对于业务流 i , 首先计算出总的丢弃概率 P_i , 在第 N 个滑动时间间隔内 N 数据包的丢弃概率为 P_{i_N} , OUT 数据包的丢弃概率为 P_{i_OUT} . 应先丢弃 OUT 的数据包, 使尽可能多的 N 数据包进入路由器中.

3.4 核心 / 出口边界路由器

图 3 展示了在核心 / 出口边缘路由器中对于 FIFO 输出链路队列的管理算法

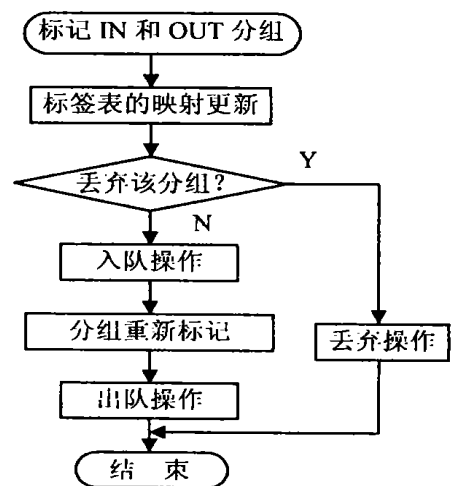


图 3 输出队列管理算法

路由器根据到达业务流中数据包头部中的标记将它映射为聚集流 AG . 映射的基本过程是将业务流的公平索引值 f_{i_N} 和 f_{i_OUT} 变成聚集流的公平索引值 F_{j_N} 和 F_{j_OUT} . 那么有下式成立:

$$F_j = F_{j_N} + F_{j_OUT},$$

$$R_j(N) = (A_{j_N}(N) + A_{j_OUT}(N)) / F_j \quad (8)$$

到达的业务流会根据丢弃概率 $P_{i_N}(N)$ 和 $P_{i_OUT}(N)$ 被丢弃。在队列中将被转发的数据包的公平索引值标记更新为

$$F_{i_N}(N) = F_{i_N}(N) \times (1 - P_{i_N}(N)),$$

$$F_{i_OUT}(N) = F_{i_OUT}(N) \times (1 - P_{i_OUT}(N)).$$

(9)

4 仿真与分析

为验证所提出模型,在Linux下扩展NS^[9]的DiffServ模块,建立一个结合DiffServ模型的加权公平主动队列管理模块。该仿真系统能够验证边缘路由器对数据包分类、标记等功能,同时又能通过对各个业务流的吞吐量来验证核心路由器上队列管理算法是否能够保证不同数据流的公平性。

通过仿真实验来验证队列管理算法公平性的仿真拓扑图如图4所示。Edge节点用来发送业务流, Sink节点用来接收业务流,两节点间的链路带宽为1.5Mbps。

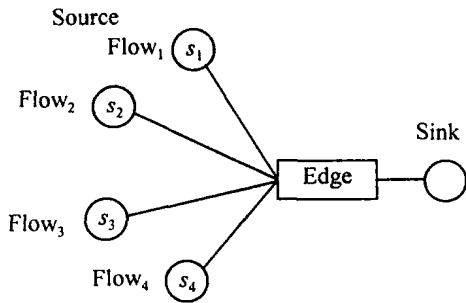


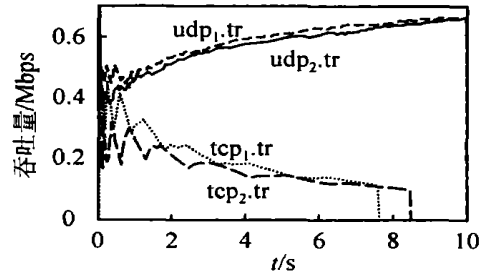
图4 仿真拓扑

针对队列管理对比DropTail和加权公平队列管理两种情况下的公平性进行仿真实验。

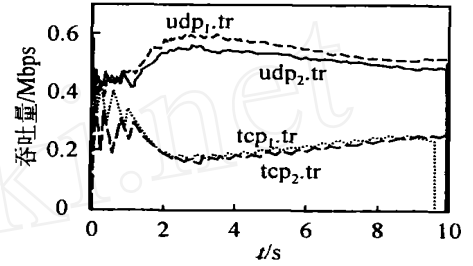
实验1 在节点Edge上采用DropTail队列管理和FIFO调度策略。节点Edge上有2个基于UDP的CBR业务流和2个基于TCP的FTP业务流,各自的速率分别为0.5Mbps, 0.5Mbps, 1Mbps, 1Mbps。这4个业务流在0时刻同时发送数据,在仿真结束时刻(10s)结束发送。

实验2 在节点Edge上采用加权公平队列管理算法和FIFO调度策略。对4个业务流的轮廓速率配置分别为: 0.25Mbps, 0.25Mbps, 0.25Mbps, 0.25Mbps。因为业务流1和2的发送速率为0.5Mbps,所以可以归为一组,同样业务流3和4也可以归为一组。这样,采用加权公平队列管理算法后,业务流聚合为两组。

两种不同仿真情形下,在Sink节点得到网络的吞吐量如图5所示。



(a) RED 算法



(b) 公平算法

图5 混合业务流的吞吐量

由图5(a)可以看出,由于拥塞窗口的作用,两个基于TCP的FTP响应流的吞吐量在仿真开始后呈下降趋势,在7.8s和8.3s左右,因为竞争不过UDP非响应流,其吞吐量接近为0。由于没有采用拥塞控制机制,两个基于UDP的CBR非响应流的吞吐量从仿真开始到结束一直呈上升趋势,链路的大部分带宽被非响应流所占据。

由图5(b)可以看出,采用加权公平队列管理算法后,根据式(1)和(4),当UDP流的速率超过TCP流时,其丢弃概率也大于TCP流,从而受到相应的惩罚,使基于TCP的响应流得到了良好的保护,公平性较实验1有较大提高。虽然还不能在混合业务流间实现完全的公平性,但考虑到算法不占用路由器更多的资源,能获得这样的性能已非常难得。

5 结语

DiffServ网络具有较好的区分服务功能,配置不同级别的业务流能够得到不同级别的服务。本文通过将加权公平主动队列管理算法部署到DiffServ模型中,实现在占用较少系统资源的前提下提高响应流与非响应流之间和不同响应流之间的公平性。

参考文献(References)

[1] 林闯,单志广,任丰原. 计算机网络的服务质量[M]. 北京:清华大学出版社,2004:26-50,74-102.
(Lin C, Shan Z G, Ren F Y. Quality of Service of Computer Networks [M]. Beijing: Press of Tsinghua University, 2004: 26-50, 74-102.)
[2] Anjum F M, Tassiulas L. Fair Bandwidth Sharing

- Among Adaptive and Non-adaptive Flows in the Internet [A]. *IEEE Infocom '99* [C]. New York: IEEE Press, 1999
- [3] Sang A M, Zhu H F, Li S Q. Weighted Fairness Guarantee for Scalable Differentiated Services (DiffServ) [A]. *ACM SigComm 98* [C]. Carlson, 1998
- [4] Dario Bauso, Laura Giarré Giovanni Neglia. AQM Stability in Multiple Bottleneck Networks [J]. *ICC 2004-IEEE Int Conf on Communications*, 2004, 27(1): 2267-2271.
- [5] Fernando Paganini, Zhikui Wang, John C Doyle, et al. Congestion Control for High Performance Stability, and Fairness in General Networks [J]. *IEEE/ACM Trans on Networking*, 2005, 13(1): 43-56
- [6] Liu W, Yang Z K, He J H, et al. Analysis and Improvement on the Robustness of AQM in DiffServ Networks [J]. *ICC 2004 IEEE Int Conf on Communications*, 2004, 27(1): 2297-2301.
- [7] Madhusudan Hosagrahara, Harish Sethu. Maximum Fairness in Input Queued Switches [A]. *Proc of ACM Sigcomm* [C]. New York: ACM Press, 2005
- [8] Xia L, Wang Y, Guan S Y, et al. A Scalable Approach for Fairness in MPLS Networks [A]. *IEEE ICCAS 05* [C]. HongKong, 2005: 513-517.
- [9] The Network Simulator (NS) HomePage [DB/OL]. <http://www.isi.edu/nsnam/ns/index.html>, 2003

(上接第 1044 页)

参考文献 (References)

- [1] Floyd S, Jacobson V. Random Early Detection Gateways for Congestion Avoidance [J]. *IEEE/ACM Trans on Networking*, 1993, 1(4): 397-413
- [2] Athuraliya S, Li V H, Low S H, et al. REM: Active Queue Management [J]. *IEEE Network*, 2001, 4(6): 236-252
- [3] Manfredi S, Bernardo M D, Garofalo F. A Robust Approach to Active Queue Management Control in Networks [A]. *Proc of Int Symposium on Circuits and Systems* [C]. Vancouver, 2004: 23-26
- [4] Kourosh Rahnamai, Payman Arabshahi, Andrew Gray. Model Predictive Neural Control of TCP Flow in AQM Networks [EB/OL]. <http://payman.caltech.edu/04-3.pdf>, 2004-12-15
- [5] Howard Demuth, Mark Beale. *Neural Network Toolbox User's Guide* [M]. Cambridge: The MathWorks Inc, 2002
- [6] Hagan M T, Menhaj M. Training Feedforward Networks with the Marquardt Algorithm [J]. *IEEE Trans on Neural Networks*, 1994, 5(6): 989-993
- [7] Yu Y B, Cao C X, Yu G Y, et al. Design of Neural Model Predictive Controller for Active Queue Management [A]. *Proc of 2005 Int Conf on Machine Learning and Cybernetics (ICMLC 2005)* [C]. Guangzhou, 2005: 1412-1416

下 期 要 目

- 反推自适应控制的发展及应用 董文瀚, 等
- 基于复杂过程简化模型的DHP 学习控制 陈宗海, 等
- 基于循环经济的开放动态投入产出智能决策系统 张文红, 等
- 基于动态Snake 模型的机械手运动轨迹视觉跟踪 吕遐东, 等
- 动态加权最小二乘支持向量机 范玉刚, 等
- 钢铁生产中的加热炉优化调度算法研究 宁树实, 等
- 基于共享最近邻聚类和模糊集理论的分类器 李订芳, 等
- 彩色地图分色算法的改进及实现 谢建华, 等
- TCSC 控制的一种新自适应Backstepping 方法 付俊, 等