分组公平调度算法的性能研究

马 争, 巢 剑
(电子科技大学通信与信息工程学院, 四川成都 610054)

摘 要: 本文针对不同的分组公平类调度算法(PFQ, Packet Fair Queuing), 对它们在端到端的时延、时延抖动、公平性等服务质量(QoS)方面作了比较, 并给出两种可以减少时标比较次数的方法, 以简化它们在硬件上实现的复杂度.

关键词: 分组公平算法; 时标; GPS
中图分类号: TN915. 05 文献标识码: A 文章编号: 0372-2112 (2003) 10-1555-03

A Study on the Performance of the PFQ Algorithms

MA Zheng, CHAO Jian
(College of Communication and Information Engineering, UESTC, Chengdu, Sichuan 610054, China)

Abstract: This paper mainly compares some different PFQ scheduling algorithms according to their QoS parameters such as end-to-end delay, delay jitter and so on. Two different methods to reduce complexity of these algorithms in hardware implementation are also provided.

Key words: packet fair queuing; timestamp; GPS

1 引言

目前高速网络的调度算法层出不穷, 自 A. Demers 等首次提出了一种适用于 Internet 网关的 IP 分组公平排队算法后, 就有很多学者不断地继续深入研究公平排队算法. 这里所谓的公平是指可以使各个连接在时延、吞吐量等方面比较一致, 有效地防止恶意用户滥用资源. 对于调度算法, 尤其是分组公平类调度算法, 怎样衡量它的有效性和可实现性, 本文给出了在一定约束条件下的各算法之间在时延、公平性以及算法复杂度等方面的大致比较.

2 分组公平类调度算法

传统的分组公平调度算法均来源于通用服务器共享(GPS)模型, 此模型是指系统中所有业务可以同时占用服务器资源, 每条业务流占用相应比例服务器的资源, 这样可以对不同业务流实现隔离, 公平地进行调度[1]. GPS 有两点假设: (1) 分组无限可分(流体模型); (2) 系统能够同时对多个连接(session)进行服务(共享服务器). 这样, GPS 既可以保证各连接的最小带宽, 又能绝对公平地在各个连接之间分配带宽, 它是属于 work-conserving 的算法. 当连接(session)的参数由漏桶加以限制后, GPS 系统能够提供最坏情况下的排队时延保证(worst-case queuing delay guarantees), 并且这个时延只与连接本身的参数相关, 而与其他连接分组到来的情况无关. 流体 GPS 算法对综合服务网络(IntServ networks)来说有理想的特性, 然而根据 GPS 模型的两点假设就可以知道, 它是无法实现的.

因此, 根据对 GPS 模型的研究中, 衍生出一种基于虚拟时间(virtual time)实现的思想, 即在每一个分组到达的时刻对其赋予一个虚拟开始或完成时间标签(timestamp), 按照虚拟时间标签的增序对分组逐一进行服务. 所谓系统虚拟时间函数, 是一个随时间变化的函数, 用来模拟 GPS 系统对所有连接的服务情况. 时标则是赋以分组的一个系统虚拟时间值, 包括分组开始时标, 分组结束时标, 其含义分别是分组在 GPS 系统中用虚拟时间表示的开始享受服务的时间和结束服务的时间. 与 GPS 相关算法的实现代价主要由下列三点所决定:

(1) 计算系统势能函数(system-potential function) (即虚拟时间函数) 的复杂度;

(2) 为选择所要发送具有最小时标的信元而进行时标分类的复杂度;

(3) 处理和存储时标的开销.

加权公平排队(WFQ)、最差情形公平加权公平排队(WF^2Q)与 GPS 算法的虚拟时间函数相同, 如下:

$$V(t)=\int_0^t \frac{1}{\sum_{i\in B(S)} \phi_i}\,dS \tag{1}$$

而改进的最差情形公平加权公平排队(WF^2Q+)算法(本质与 sSPFQ, shaped Starting Potential Fair Queuing 即整形的启始势公平算法, 是相同的), 它的虚拟时间函数则为:

$$V_{WF^2Q+}(t+S)=\max\left\{V_{WF^2Q+}(t)+W(t,t+S),\ \min_{i\in B(t)}\{S_i^{h_i(t)}\}\right\} \tag{2}$$

没有阻塞时, 整形的虚拟时钟(shVC, shaped Virtual Clock)

收稿日期: 2002-06-27; 修回日期: 2003-05-30

算法采用实时时间为其虚拟时间; 在有阻塞的情况下, 由于 shaper(整形器) 的作用, 分组的虚拟发送时间为实时时间加上在队列中的等待时间, 这简化了算法的复杂度, 但带来了不公平性和较大的时延. 自时钟公平(SCFQ, Self2Clock Fairing Queu2ing) 算法的虚拟时间直接为上一个分组的虚拟结束时标, 但它的一个改进算法))) 最小时延的自时钟公平(MD2SCFQ, Minimized Delay2SCFQ) 算法[4] 的虚拟时间函数计算较为不同, 如下:

$$V(S_i)=\frac{\sum_{i\in B(S_j)} F_i r_i-\sum_{i\in B(S_j)} l_i}{\sum_{i\in B(S_j)} r_i} \tag{3}$$

对于大多数从 GPS 模型衍生出的分组公平调度算法来说, 它们的开始和结束时标基本上都是一样的:

$$S_i^k=\max\{V(a_i^k),F_i^{k-1}\} \tag{4}$$

$$F_i^k=S_i^k+(L_i^k/r_i) \tag{5}$$

从理论上看, WF^2Q 是比较好的一种算法, 它能精确地跟踪在 GPS 中处于活动的连接以计算分组的虚拟时间, 但是随着传输速度的提高以及链路数的增加, 其算法的复杂度(虚拟时间的计算和被标记过的分组的分类排序) 急剧上升, 这使得它很难适应高速网络的要求. WF^2Q 是一种不可实现的算法, WF^2Q+ 是 WF^2Q 的一种改进算法. 它同 WF^2Q 一样具有公平性和时延的保证, 比较理想地接近 GPS 算法模型, 只是比 WF^2Q 系统虚拟时间函数的计算的复杂度要降低了很多(见表 1) . 然而, WF^2Q+ 与 WF^2Q 一样只能支持少量的连接速率, 当连接数增加时, 其复杂度呈线性增长.

不同的 PFQ 算法关键点在于它们对分组的虚拟时间的计算方法和对队列中分组的选择策略不同而不同, 一般的分组选择策略包括: 最小虚拟开始时间优先(Smallest virtual Start time First, SSF) 、最小虚拟完成时间优先(Smallest virtual Finish time First, SFF) 以及最小合格虚拟完成时间优先(Smallest Eligi2ble virtual Finish time First, SEFF) . 前两种策略对分组的选择都是只用一个时标, SSF 使用开始时标, SFF 使用结束时标. 使用一个时标并不能很好的逼近 GPS 模型, 而 SEFF 方法的不同之处在于它不止使用一个时标, 它是在所有合格的分组中进行选择, 这里的合格指分组的虚拟开始时间不大于当前的系统虚拟时间.

在研究与 GPS 相关的 PFQ 算法模型过程中, 许多 PFQ 类算法被相继提出, 表 1 给出了几种 PFQ 算法在一定约束情况下时延、公平性等方面的对比. 其中 MD2SCFQ 和 WF2Q+ 既保证了较好的时延, 又提供了很好的公平性.

表 1 几种分组公平调度算法的比较

算法名称	分组选取原则	时 延	公 平 性		虚拟时间函数的复杂度
			SFI	WFI	
GPS	流体, 无	$\frac{D_i}{r_i}$	0(最好)	0(最好)	O(N)
WFQ	SFF	$\frac{D_i}{r_i}+\frac{L_{max}}{C}$	$\max\left(\frac{L_i}{r_i}+\frac{L_{max}}{r_j}+C_i,\frac{L_j}{r_j}+\frac{L_{max}}{r_i}+C_j\right)$	O(N)	O(N)
WF^2Q	SEFF	$\frac{D_i}{r_i}+(N-1)\frac{L_{max}}{C}$	$\frac{L_{max}}{r}$	$L_{i,max}+(L_{max}-L_{i,max})\frac{r_i}{r}$	O(N)
WF^2Q (= ssPFQ)	SEFF	同上	$2\max\left(\frac{L_i}{r_i},\frac{L_j}{r_j}\right)+\max\left(\frac{L_{max}}{r_i},\frac{L_{max}}{r_j}\right)$	$L_{i,max}+(L_{max}-L_{i,max})\frac{r_j}{r}$	O(logN)
shVC	SEFF	同上	很差	很差	O(1)
SCFQ	SFF	同上	$\frac{L_i}{r_i}+\frac{L_j}{r_j}$	O(N)	O(logN)
SFQ	SSF	同上	同上	一般	O(1)
MD2SCFQ	SEFF	$\frac{D_i}{r_i}+\frac{L_{max}}{C}$	$\max(f_{i,j},f_{j,i})$	较好	同上
FFQ	SFF	同上	$2T+\max\left(\frac{L_i}{r_i},\frac{L_j}{r_j}\right)$	一般	同上
SPFQ	SFF	同上	$\max\left(\frac{L_i}{r_i},\frac{L_j}{r_j}\right)+\max_{1\le k\le N}\left(\frac{L_k}{r_k}+\frac{L_{max}}{r}\right)$	一般	O(logN)

注: C_i 端到端的时延受漏桶(D_i, r_i) 约束

$C_i=\min\left((N-1)\frac{L_{max}}{r_i},\max_{1\le k\le N}\frac{L_k}{r_k}\right)$ $f_{i,j}=\frac{L_i}{r_i}+\max\left(\frac{L_{max}}{r_j},\max_{1\le k\le N}\frac{L_k}{r_k}-\frac{r_i}{r-r_j}\left(\max_{1\le k\le N}\frac{L_k}{r_k}-\frac{L_i}{r_i}\right)-\frac{L_i}{r}\right)$

i: 连接号; L: 分组长度; C: 链路速率; r: 速率; N: 连接数; D: 漏桶容量; T: 帧间隔

3　降低时标比较数的两种方法

既然分组公平算法的基本思想就是给队列中每个到达的分组赋予一个时标, 然后基于时标进行计算和比较操作, 所以其复杂度就在于如何减少时标计算的复杂性和分类、比较的次数. 在 GPS 类算法当中, 能够很好的获得时延特性, 最大降低硬件实现复杂度的算法当中, 目前存在两种主要的分类方法: 离散速率调度器(discrete2rate scheduler)和对数日历队列(Logarithmic Calendar Queue, LCQ). LCQ 是一种最优的日历分类方法, 它使用一种理想的方法, 通过增加用于分类时标的二进制的粒度以减小复杂度, 这样, 对于每条连接来说, 时延范围的相对降低被最小化了. Grouping(分群)方法是一种很好的减少时标比较数的方法, pe2rate grouping(按每速率分群)[2]和 pe2service interval grouping(按每服务间隔分群)[3]就是其中两个. 在 pe2rate grouping 方法中, 具有相同预约速率的连接被放在同一个群之中, 群中存储着每条处于活动的连接的入口, 入口包含一个指向连接队列头的指针和队列头信元的虚拟起始时间, 具有最小虚拟起始时间的连接被放入调度器中以调度, 如图 1 所示. 这样, 由于每条连接内部不需要进行时标比较, 而只需要比较连接的头信元, 参加比较的时标数由原来的每个信元降到每条连接的头信元, 大大降低了算法的复杂度.

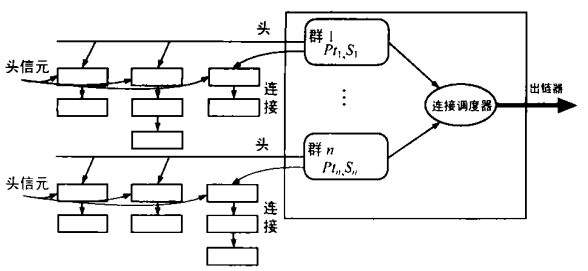

图 1　pe2rate grouping 方法的结构方式

然而, pe2rate grouping 方法只适用于定长分组的环境中, 如 ATM, 对于变长分组的调度则不适用. pe2service interval grouping 则是一种可以适用于变长分组环境中的方法. 此方法将服务间隔(Service Interval)相同的分组放在同一个服务间隔组里, 在不同的服务间隔组之间进行时标的比较以确定下一个要发送的分组. 同一个服务间隔组内的分组则不需要比较排序, 而是根据分组的结束时标计算分组在一个日历队列中的位置, 将分组插入进去. 这也大大减少了分组的比较次数, 在一定程度上降低了算法的复杂度.

以上两种减少时标比较数的方法都使得算法实现的复杂度降低为系统所支持的连接数上, 从而避免了大量运算所消耗的硬件资源, 具有一定的可行性.

4　小结

可以看出, 对于分组公平类调度算法是一类能够在时延和公平性方面比较好的调度算法. 在众多的分组公平类调度算法中, 最为简单的算法是 shVC, 在实际应用中, 它经常用于定长分组环境中的调度. 但在能够提供公平性和时延的情况下, WF^2Q+ 和 MD2SCFQ 是比较好的选择, 但在实现上由于其算法本身的复杂度可以考虑在实现时进行一定程度的简化.

参考文献:

[1] J C R Bennett, D C Stephens, Hui Zhang. High Speed, Scalable, and Accurate Implementation of Packet Fair Queueing Algorithms in ATM Networks[A]. Network Protocols, 1997. Proceedings, 1997 International Conference on[C]. San Francisco, CA, USA 1997. 7- 14.

[2] J C R Benett, Hui Zhang. Hierarchical packet fair queuing algorithms [J]. IEEE Trans on Networking, 1997, 5(5): 675- 689.

[3] D C Stephens, J C R Bennett, Hui Zhang. Implementing scheduling algorithms in high2speed networks[J]. IEEE JSAC, Special Issue on Next Generation IP Switches and Routers, 1999, 17(6): 1145- 1158.

[4] F M Chiussi, A Francini. Minimum2Delay Self2Clocked Fair Queueing Algorithm for Packet2Switched Networks[A]. IEEE Infocom98[C]. Holmdel, NJ, USA, 1998. Ó- 1112- 1121.

作者简介:

马　争　男, 1957 年生于南京, 1982 年 2 月毕业于成都电讯工程学院, 现为电子科技大学中山学院院长, 教授, 中国电子学会、四川电视分会专家委员会委员, 电子工业出版社计算机分委会委员, 主要从事计算机网络技术、计算机应用、计算机安全、智能通信网、图像与语音信号处理的研究工作, 获得多项省部级奖励, 出版专著四本, 发表论文 30 余篇, 目前主要研究方向在视频压缩、数字移动通信、计算机网络应用、网络安全等方面.

巢　剑　男, 1976 年生于江西省九江市, 1999 年和 2002 年在电子科技大学通信与信息工程学院获学士、硕士学位, 曾在南京富士通通信股份有限公司和深圳华为技术有限公司从事通信软硬件的设计与开发, 目前在深圳电信深大电话有限公司工作, 主要从事网络规划与安全, 及数据库开发方面的工作.