一种基于紧急程度的自时钟开始时间公平排队分组调度算法

刘文波 郭云飞 马海龙
(国家数字交换系统工程技术研究中心 郑州 450002)

摘 要：为了克服目前 GPS (Generalized Processor Sharing)类调度算法中实时应用分组的排队时延较大且不稳定的局限性，该文提出一种新的分组排队调度算法，该调度算法在计算分组服务标签时添加了一个紧急程度函数，调整了到达分组间的竞争关系，从而可以按照实时性应用的要求来调整到达分组的转发先优级，由此显著降低了实时性应用分组的排队时延和抖动幅度。分析和仿真实验表明，与 GPS 类其它调度算法相比，该调度算法对于实时应用的分组能提供较低的、更稳定的排队时延保证，同时还继承了 GPS 类算法的公平性和排队时延有界等特性，而且对系统虚拟时间的跟踪计算更为简捷高效。

关键词：分组排队调度；紧急程度函数；排队时延；公平性；系统虚拟时间
中图分类号：TP393 **文献标识码**： A **文章编号**：1009-5896(2010)06-1452-05
DOI: 10.3724/SP.J.1146.2009.00781

An Urgency Based Self-clock Start-Time Fair Queuing Packet Scheduling

Liu Wen-bo Guo Yun-fei Ma Hai-long
(*National Digital Switching System Engineering & Technological R&D Center*, *Zhengzhou* 450002, *China*)

Abstract: In order to get over the limitation of the GPS(Generalized Processor Sharing) like queuing algorithms which the bound of the queuing delay is so long and unstable that is undesirable for the packets of real-time applications, a new packet queuing and scheduling algorithm is proposed. In this algorithm the competitions of the arrival packets are fine-tuned by means of adding a function of flow urgency degree on the computation of their service tags, so the transmitting priorities of the arrival packets can be tuned according to their real-time applications' needs, therefore the queuing delay and the bounds of oscillation are cut down significantly. According as the analysis and the simulation, this packet queuing and scheduling algorithm can resolve the long and unstable packet latencies problem for real-time applications, and shares both the bounded-delay and fairness properties of the GPS like algorithms, and the computation of its system virtual time is simple and efficient.

Key words: Packet queuing and scheduling; Function of urgency degree; Queuing delay; Fairness; System virtual time

1 引言

分组排队调度是分组交换网络提供 QoS (Quality of Service)保证的关键，相关研究很多，保证公平性的调度算法主要有两类：GPS[1]类和轮询类。GPS 类的代表性调度算法有 PGPS[1](Packet by packet Generalized Processor Sharing)，WF2Q[2] (Worst-case Fair weighted Fair Queuing)，SCFQ[3] (Self-Clocked Fair Queuing)和 SFQ[4](Start-time Fair Queuing)等，轮询类的代表性调度算法有 DRR[5](Deficit Round Robin)，NDRR[6](Nested Deficit Round Robin)等。这些调度算法虽然都能保证调度的公平性和分组的排队时延，但共同弱点是时延保证范围的阈值较大，且不够稳定。在不破坏 GPS 类调度算法调度公平性的前提下，本文针对如何保证实时业务流分组排队的时延范围和稳定性问题进行了研究。

理想化的排队调度算法是由 Parekh 等人在文献[1]中提出的 GPS 调度算法。在 GPS 算法中，假定每个流都是可以进行无限细分的液态流，调度器能够同时并发地服务多个流，在调度时，只要有业务流到达调度器就可以按照带宽预留的比例对到达的流量进行调度服务。而在实际中，分组是不可细分的，且必须串行地进行调度，因此 GPS 算法不能直接用于分组排队调度。为了将 GPS 算法的思想用于分组排队调度，前人提出了很多模仿 GPS 算法的分组排队调度算法，其中，最具代表性的是 PGPS 调度算法。在总体吞吐能力相等的要求下，作为 GPS 算法的一种模仿，由于分组到达时间和分组长度的不确定性，使得 PGPS 调度器无法完全与 GPS 调度器的分组调度顺序完全一致，在文献[2]中，Bennett 等人证明了一个分组离开 PGPS 调度

2009-05-22 收到，2009-10-08 改回
国家 973 计划项目(2007CB307102)和国家 863 计划重大项目(2008AA01A323)资助课题
通信作者：刘文波 lwb@mail.ndsc.com.cn

器的时间有可能要比其离开对应 GPS 调度器的时间要早得多，因此个别流的分组突发性和抖动程度可能会很大，这对实时性的流媒体类应用来说是不希望见到的，为此，Bennett 等人提出了 WF2Q 算法。然而，实现 WF2Q 需要精确地跟踪计算其对应 GPS 系统中的虚拟时间。Zhao 等人在文献[7]中深入分析了 WF2Q 及其它 GPS 类调度算法中精确跟踪计算 GPS 虚拟时钟的计算复杂度和实现代价，证明了跟踪计算 GPS 虚拟时钟时的计算复杂度为 $O(n)$，n 为调度器中已到达的分组个数，所以 PGPS 和 WF2Q 的计算复杂度很高，在具体实现时较复杂。因此，SCFQ 和 SFQ 等调度算法中不对 GPS 虚拟时钟进行精确跟踪计算，而是使用 $O(1)$的代价维持一个粗略的系统内部时钟，例如，都将 t 时刻对应的系统内部时钟 $V(t)$ 定义为 t 时刻调度器正在服务的分组的开始时间标签。这种改进虽然会大大消减跟踪计算系统时钟的复杂度，降低了系统实现的复杂性，但是除了会导致调度算法的公平性和排队时延略微不如 PGPS 或 WF2Q 以外，还会过多地出现实时流分组的结束时间标签或开始时间标签与其它流的多个到达分组的结束时间标签或开始时间标签相等的情况，导致实时流的分组必须要与其它类型流的到达分组平等竞争，从而引起实时流的分组排队时延更大和不平稳。Gebali 在文献[8]中对这种情况进行了理论分析，Akesson 等人在文献[9]中以及 Bredel 等人在文献[10]中对这种情况的影响进行了研究并提出了一些改进策略。

为了克服目前 GPS 类调度算法中实时流分组的排队时延较大且不稳定的局限性，本文在文献[9,10]的基础之上，通过在计算服务标签时添加紧急程度函数，以紧急程度函数来精细调整到达分组的竞争关系的办法，提出了一种基于紧急程度的自时钟开始时间公平排队(Urgency Based Self-clock Start-time Fair Queuing，UBSSFQ)调度算法。分析和实验表明，与其它 GPS 类调度算法相比，UBSSFQ 调度算法在继承 GPS 类算法的公平性和排队时延有界的同时，对于实时应用的分组还能提供较低的、更稳定的排队时延，而且系统虚拟时钟的跟踪计算较为简捷高效。

2　UBSSFQ 调度算法

本文提出的 UBSSFQ 算法具体定义如下：

(1)系统虚拟时间的初始化与计算。开始时系统虚拟时间设为 0，在调度器忙时，t 时刻对应的系统虚拟时钟 $V(t)$ 定义为

$$V(t)=\max\{S(t),V(t^-)\} \tag{1}$$

其中，$V(t^-)$ 是 t 时刻之前计算出的系统虚拟时间，$S(t)$ 是 t 时刻调度器正在服务的分组的开始时间标签。

(2)分组开始时间标签的计算。分组 P_i^k 到达时，开始时间标签 S_i^k 通过以下函数进行计算：

$$S_i^k=\max\{F_i^{k-1},V(A_i^k)-\text{UrgencyDegree}()\} \tag{2}$$

其中，P_i^k 表示第 i 个流的第 k 个分组，A_i^k 表示 P_i^k 的到达时刻，$V(A_i^k)$ 表示 A_i^k 时刻对应的系统虚拟时间，S_i^k 表示 P_i^k 的开始时间标签，F_i^k 表示 P_i^k 的结束时间标签。F_i^k 定义如下：

$$F_i^k=S_i^k+L_i^k/r_i \tag{3}$$

其中，F_i^0 =0，L_i^k 是 P_i^k 的分组长度，r_i 是流 i 的预留带宽。

(3)调度。依据分组开始时间标签的递增顺序进行分组调度，在所有到达分组中选取一个有最小开始时间标签的分组进行服务。

(4)调度器空闲时的重新初始。调度器一旦空闲则重新初始，系统虚拟时间 $V(t)$ 设为 0。

UBSSFQ 在计算分组的时间标签时增添了一个 UrgencyDegree( )函数，正由于这个函数，当一个实时流分组到达时，使得其分组的开始时间标签将适当小于其它流分组的开始时间标签，实时流的分组将会得以优先传递，所以把函数 UrgencyDegree( )的值称为紧急程度。根据 Golestani 在文献[2]中和 Goyal 在文献[4]中的分析，为了达到与 SCFQ 相同的公平性，本算法中定义的 UrgencyDegree( )的值应不大于 $L_{\min}/r$，其中，$L_{\min}$ 是最小的分组长度，r 是输出带宽。

对于实时流来说，在保证公平性的前提下，为了让一个实时流分组得到优先传递，应使得该流的紧急程度尽可能是一个较大值，但是其它流也可能有一个较大的紧急程度。因此，实时流有可能受到竞争。所以，在 UBSSFQ 中，流的紧急程度要按照协议或管理的需要进行等级划分，将不同流按照不同的紧急程度等级归入不同的类。

在理想情况下，实时应用的业务流是恒定比特率的，实时分组产生的时间间隔相同，同时，不耗尽所分配到的预留带宽；所以实时业务流的队列可能时常变为空，因此，当一个实时分组来到时，对应的实时流通常是一个新的流。在这种情况下，我们可简单地将紧急程度设为一个无限小的常量。因为在恒定的分组到达间隔期间只有一个分组，那么这个无限小的常量就足以使实时流分组的开始时间标签比其它与其竞争的分组的小，因此能够得到快速且平稳的调度转发。

在实际网络中，如上所述，分组的到达时间间隔可能不同，实时业务流的队列有时有可能空，有

时也可能有一些分组以紧凑的方式到达，无限小常量值的紧急程度对于那些紧凑到达的分组来说是不够的，在不破坏公平性的前提下，需要将 Urgency Degree ()设定为

$$\text{UrgencyDegree}(\text{flow}_i)=\frac{L_{\min}}{r}\times d_{\text{urgency}} \tag{4}$$

其中将式(4)中的 $L_{\min}$ 设定为网络中的最小包长，d_{urgency} 表示紧急等级，取值在[0,1)之间。当实时流出现分组紧凑到达的情况时，在不破坏公平性的前提下，通过紧急程度值的调节，比其它流分组的开始时间标签小的紧凑到达的实时流分组会比较多，在调度转发时，多个紧凑到达的实时流分组将有较高的转发优先级，使得实时流分组尽可能地得以优先转发。所以，紧急程度值是在这种情况下使得实时流分组能够首先转发的主要因素。然而，当有很多不同实时性要求的实时流时，就需要有多个不同的 d_{urgency} 来指示不同的紧急程度。

3 公平性分析

为证明 UBSSFQ 是公平的，需要证明对于任意 $(t_1,t_2]$ 时间区间内的两个活动的流 f 和 m 有 $\left|\frac{W_f(t_1,t_2)}{r_f}-\frac{W_m(t_1,t_2)}{r_m}\right|$ 有界，其中，$W_i(t_1,t_2)$ 是流 i 在 $(t_1,t_2]$ 时间区间内得到的服务量，r_i 是流 i 分配到的带宽。为此，首先来分析一下 $W_i(t_1,t_2)$ 的上界和下界。

引理 1 如果流 i 在 $(t_1,t_2]$ 时间区间一直是活动的，则在 UBSSFQ 中有式(5)成立，

$$r_i(v_2-v_1)-L_i^{\max}\le W_i(t_1,t_2) \tag{5}$$

其中 $v_1=V(t_1), v_2=V(t_2)$，$L_i^{\max}$ 是流 i 中最大分组长度。

证明 因为 $W_i(t_1,t_2)\ge 0$，所以如果 $r_i(v_2-v_1)-L_i^{\max}\le 0$，则式(5)显然成立。所以只需考虑 $r_i(v_2-v_1)-L_i^{\max}\ge 0$ 的情况，即 $v_2\ge v_1+L_i^{\max}/r_i$ 的情况。

假定 P_i^1 是流 i 的第 1 个分组，分析流 i 在 $(v_1,v_2]$ 区间内得到的最大服务量，考虑下面两种情况：

(1)存在一个分组 P_i^n，使得 $S_i^n<v_1$ 且 $F_i^n>v_1$ 的情况：因为流 i 在 $(t_1,t_2]$ 时间内是活动的，所以 $V(A_i^{n+1})\le v_1$；所以从式(2)，式(3)和式(4)可知 $S_i^{n+1}=F_i^n$；又因为 $F_i^n\le S_i^n+L_i^{\max}/r_i$ 且 $S_i^n<v_1$，所以有 $S_i^{n+1}<v_1+L_i^{\max}/r_i<v_2$；而 $S_i^{n+1}=F_i^n$ 且 $F_i^n>v_1$，所以有 $S_i^{n+1}\in(v_1,v_2]$。

(2)存在一个分组 P_i^n，使得 $S_i^n=v_1$ 的情况：因为流 i 在 $(t_1,t_2]$ 时间内是活动的，因此 $V(A_i^{n+1})\le v_1$，所以 $S_i^{n+1}=F_i^n$；又因为 $F_i^n\le S_i^n+L_i^{\max}/r_i$ 且 $S_i^n=v_1$，所以有 $S_i^{n+1}\le v_1+L_i^{\max}/r_i<v_2$；而 $S_i^{n+1}=F_i^n$ 且 $F_i^n>v_1$，所以有 $S_i^{n+1}\in(v_1,v_2]$。

所以存在满足 $S_i^k\in(v_1,v_2]$ 的分组 p_i^k，且有 $S_i^k\le v_1+L_i^{\max}/r_i$。

假设 P_i^{k+m} 是在 $(v_1,v_2]$ 虚拟时间区间内得到服务的最后一个分组，即 $F_i^{k+m}\ge v_2$，则因为 $S_i^k\le v_1+L_i^{\max}/r_i$，所以有 $F_i^{k+m}-S_i^k\ge(v_2-v_1)-L_i^{\max}/r_i$。

而流 i 在 $(v_1,v_2]$ 区间内一直是活动的，所以有 $F_i^{k+m}=S_i^k+\sum_{j=0}^{m}\frac{L_i^{k+j}}{r_i}$，即 $\sum_{j=0}^{m}\frac{L_i^{k+j}}{r_i}=F_i^{k+m}-S_i^k$；由此 $\sum_{j=0}^{m}\frac{L_i^{k+j}}{r_i}\ge(v_2-v_1)-\frac{L_i^{\max}}{r_i}$，所以有 $\sum_{j=0}^{m}L_i^{k+j}\ge r_i(v_2-v_1)-L_i^{\max}$ 成立。

因为 $S_i^{k+m}\le v_2$，分组 P_i^{k+m} 到 t_2 时刻服务完毕，所以 $W_i(t_1,t_2)\ge\sum_{j=0}^{m}L_i^{k+j}$。 证毕

引理 2 在 UBSSFQ 调度器中，对于任意流 i 在 $(t_1,t_2]$ 时间区间内中有式(6)成立：

$$W_i(t_1,t_2)\le r_i(v_2-v_1)+L_i^{\max} \tag{6}$$

其中 $v_1=V(t_1)$，$v_2=V(t_2)$，$L_i^{\max}$ 是流 i 中最大分组长度。

证明 从 UBSSFQ 的定义中可知流 i 在 $(v_1,v_2]$ 区间得到服务的分组的最小开始时间标签为 v_1，最大开始时间标签为 v_2；若流 i 在 $(v_1,v_2]$ 区间得到服务的分组集合设为 D，则集合 D 可划分成两个子集 D_1 和 D_2。

(1)D_1 是由开始时间标签不小于 v_1 而且结束时间标签不大于 v_2 的分组组成，即如式(7)所示：

$$D_1=\{k\mid v_1\le S_i^k\le v_2\wedge F_i^k\le v_2\} \tag{7}$$

结合式(2)，式(3)和式(4)可知 $\sum_{k\in D_1}L_i^k\le r_i(v_2-v_1)$。

(2)D_2 是由开始时间标签最大为 v_2 而且结束时间标签大于 v_2 的分组组成，即如式(8)所示：

$$D_2=\{k\mid v_1\le S_i^k\le v_2\wedge F_i^k>v_2\} \tag{8}$$

显然，属于 D_2 集合的分组至多有一个分组，因此 $\sum_{k\in D_2}L_i^k\le L_i^{\max}$。

从(1)和(2)的分析可知：$\sum_{k\in D}L_i^k\le r_i(v_2-v_1)+L_i^{\max}$。 证毕

因为，在任何时间区间内，两个流间不公平的最大程度可能发生在一个流接收到最多的可能服务而另一个流接收最少的可能服务，因此从引理 1 和引理 2 可直接得出如下定理：

定理 在任何时间区间 $(t_1,t_2]$ 内，若流 f 和 m

是两个活动的流，则两个流在 UBSSFQ 调度器中接受到的服务之差如式(9)所示：

$$\left|\frac{W_f(t_1,t_2)}{r_f}-\frac{W_m(t_1,t_2)}{r_m}\right|\le\frac{L_f^{\max}}{r_f}+\frac{L_m^{\max}}{r_m} \tag{9}$$

从以上定理可知，UBSSFQ 的公平性指数是最佳公平性指数[2] $\frac{1}{2}\left|\frac{L_f^{\max}}{r_f}+\frac{L_m^{\max}}{r_m}\right|$ 的 2 倍(公平性指数为 0 则为完全公平)，与目前已知的几种分组排队调度算法相比具有很好的公平性。

4　仿真分析

使用NS2作为仿真工具，仿真的拓扑结构如图1所示，r_s 和 r_d 是路由器，s_i 和 d_i 分别表示源和目的结点。结点和路由器间的链路带宽是10 Mbps，且有1 ms的传输延时，瓶颈是两个路由器间的链路，只有2 Mbps带宽，且有10 ms的传输延时，调度算法只处于 r_s 路由器。

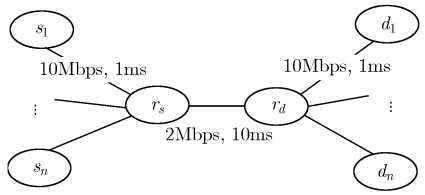

图1 仿真的拓朴图

源端发出 9 个 2 Mbps CBR 流和一个 100 kbps CBR 流，最小分组长度为 120 byte，这些流的带宽占用比例相同。其中，100 kbps CBR 的流为实时业务流，称作 flow 10，flow 10 在所有流中的优先权是最高的，紧急程度设为 0.001。

在这种情况下，flow 10的分组到达的时间间隔稍大于其它流，消耗的带宽小于预定的带宽。理想情况下，flow 10的排队延时将很小，其队列在多数情况下是空的。另一方面，如果调度算法不够理想，排队延时会很大且不稳定。

可从图2中看到这种现象，在NDRR中，当flow 10的一个分组到来时，排队的时延是由轮询份额的分配和轮询时机决定的。因此NDRR的排队时延是不稳定的。在SCFQ中，排队时延是由具有相同结束时间标签的分组数量和流的数量决定的，SCFQ的虚拟时间与精确的GPS虚拟时间相差很大，其排队时延较差。WF2Q同样有SCFQ的多个流相互竞争的问题，所以排队时延有时甚至超过NDRR的排队时延。在SFQ中，当流到达时，排队时延将只依赖于具有相同的开始时间标签的数量，其排队时延将比SCFQ平稳，但对于实时业务流来说还是不够的。在UBSSFQ中，每当flow 10的新分组到达时，分组的开始时间标签都要比其它流的时间标签小，所以flow 10的分组得以优先传递，从图2中会看到该方法具有很低且平稳的排队时延。

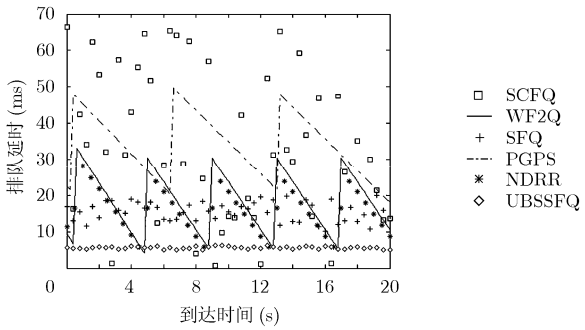

图2 实时流在仿真网络中使用不同排队调度算法的排队时延情况对比

5　结束语

本文提出了 UBSSFQ 调度算法，这是一种基于紧急程度、自定义时钟、按开始时间标签调度的分组排队调度算法。经分析和仿真实验表明，与其它 GPS 类调度算法相比，UBSSFQ 不但继承了调度公平性和时延有界的特性，并且简化了跟踪计算 GPS 虚拟时钟的计算复杂度和实现复杂性，此外，其显著优势在于：可以据算法中定义的紧急程度函数对分组的调度优先级进行调节，按照分组的实时性要求，保证实时性业务的分组有较小的排队时延。但是，UBSSFQ 仍与 GPS 类中的其它分组排队调度算法一样，对于有 n 个输入流的调度器来说，选择分组的计算复杂度为 $O(\log\ n)$，对于如何削减 UBSSFQ 中分组调度的计算复杂性还需做进一步深入的研究。

参 考 文 献

[1] Parekh A K. A generalized processor sharing approach to flow control in integrated services networks. [Ph.D. dissertation], Dept. Elec. Eng. Comput. Sci., MIT, 1992.

[2] Bennett J C and Zhang H. WF2Q: Worst-case fair weighted fair queueing. Proceedings of IEEE INFOCOM'96, San Francisco, Mar 1996, Vol. 1: 120–128.

[3] Golestani S J. A self-clocked fair queueing scheme for broadband applications. Proceedings of IEEE INFOCOM'94, Toronto, CA, April 1994: 636–646.

[4] Goyal P, Vin H M, and Chen H. Start-time fair queuing: A scheduling algorithm for integrated services. Proceedings of the ACM-SIGCOMM'96, Palo Alto, CA, August 1996: 157–168.

[5] Shreedhar M and Varghese G. Efficient fair queuing using

deficit round robin. *IEEE/ACM Transactions on Networking*, 1996, 4(3): 375–385.

[6] Kanhere S S and Sethu H. Fair, efficient and low latency packet scheduling using nested deficit round robin. Proceedings of the IEEE Workshop on High Performance Switching and Routing, Dallas, TX, May 2001: 6–10.

[7] Zhao Q and Xu J. On the computational complexity of maintaining GPS clock in packet scheduling. Proceedings of IEEE INFOCOM'04, Hong Kong, Mar. 2004, Vol. 4: 2383–2392.

[8] Gebali F. Analysis of Computer and Communication Networks. 1st edition, Berlin: Springer, 2008: 449–467.

[9] Akesson B, Steffens L, and Strooisma E, *et al*.. Real-time scheduling using credit- controlled static-priority arbitration. Proceedings of IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, Kaohsiung, Taiwan, Aug. 2008: 3–14.

[10] Bredel M and Fidler M. Understanding fairness and its impact on quality of service in IEEE 802.11. Proceedings of IEEE INFOCOM'09, Rio de Janeiro, Brazil, April. 2009: 1669–1677.

刘文波： 男，1968 年生，副教授，博士生，研究方向为宽带信息网与网络体系结构.

郭云飞： 男，1963 年生，教授，博士生导师，研究方向为宽带信息网络与信息网关防等.

马海龙： 男，1980 年生，讲师，博士生，研究方向为宽带信息网络路由算法与协议.