具有优先服务机制的嵌套式 DRR 算法

简贵胄　　葛　宁　　冯重熙
（清华大学电子工程系 北京 100084）

摘　要: 针对 DRR 算法在保证业务时延性能上的固有不足，该文提出了具有优先服务策略的嵌套式 DRR 算法，该算法对需要保证延时的业务类实施专门的服务策略，利用漏桶控制和虚令牌的分配机制，并改善服务队列的调度策略，在不增加算法复杂度的情况下，有效地减小了时延敏感业务队列中包的延迟。文章证明了算法在获得延时保证的同时，选择合适的参数，仍然能够维持调度算法的公平性。仿真结果表明算法对时延性能的改善是明显的。

关键词: 调度，DRR 算法，延时特性，公平性
中图分类号：TN913.2　　**文献标识码**：A　　**文章编号**：1009-5896(2005)01-0123-04

A Prioritized Nested DRR Algorithm

Jian Gui-zhou　　Ge Ning　　Feng Chong-xi

(*Dept of Electronic Eng., Tsinghua University, Beijing* 100084, *China*)

Abstract　As DRR fails to provide strong latency bound, in this paper,　a new scheduling discipline named Prioritized Nested DRR (PNDRR) is presented, which introduces a token bucket with virtual allocated token quantum and changes the entering order of the scheduling list for the latency-critical flow. By using the scheme for the latency critical flow, delay of packet in latency critical queue is effectively diminished. In this paper, theoretical analyses prove that PN DRR results in a significant improvement in the latency bound of delay-sensitive traffic in comparison to nested DRR(NDRR) while preserves *the good properties*　*as the same relative fairness bound and the per-packet complexity of* $O(1)$ as NDRR. Simulation results also support　analysis.

Key words　Scheduling, DRR algorithm, Delay-sensitive traffic, Fairness

1　引言

合适的调度策略对于网络支持不同的服务等级具有非常重要的意义，好的调度算法应该具有公平、有效和低复杂度等特性。近年来提出的调度算法如基于 GPS 的 WFQ（Weighted Fair Queueing）、WF^2Q(Worst-case Fair Weighted Fair Queueing)等算法[1]具有良好的延时特性，但是由于在发送包之前需要对各个队列进行排序，以决定发送的先后顺序，算法的时间复杂度为 $O(\log(n))$，因而复杂度相对较低的基于轮循策略的算法受到重视。其中 DRR（Deficit Round Robin）算法[2]能支持变长数据包的调度，且算法的时间复杂度是 $O(1)$，因而得到广泛的应用，DRR 调度算法的规则使得队列中包的延迟同队列所分配的令牌数密切相关，低速率业务分配到的令牌数少，延迟也很大。为改善 DRR 算法的延迟，提出融合公平排队思想的 DRR++算法[3]，该算法涉及到对 DRR 服务顺序的很大改动，使得调度算法的复杂度大大增加；文献[4]中提出的嵌套式 DRR 算法，简称 NDRR，采用了细粒度调度策略，可使所有被调度队列中包的延迟上界减小。在很多应用场合，低传输带宽业务对延迟要求也很小，典型的应用如 IP 电话，针对这类业务的需求，在差分服务体系架构下，将这类业务聚集到一个队列中，采用一种绝对优先级别为其服务[5]是一种方法，但是优先级别的引入破坏了原有的 DRR 算法固有的公平性。

2　改进的 DRR 调度算法

提出的算法是在 NDRR 算法基础上改进的，称为具有优先服务机制的嵌套式 DRR 算法（PNDRR）,算法将低带宽低延迟业务汇聚到一个队列中优先处理，作为优先队列，采取了下面两方面改进：首先，在调度机制上给优先队列分配以超出其实际应该得到服务量更多的令牌数 Q_C，称为“虚令牌”，同时在队列入口处，用漏桶来限制相应的业务流速率，以防止用户对网络带宽的恶意使用。这一步骤实质上解除了该队列的输出速率和所分配服务令牌数目的耦合关系。第二步，改变 NDRR 算法中进入服务链表的规则，使得优先列中

2003-07-15 收到，2003-12-11 改回
国家自然科学基金(69896242)资助课题

的包尽早获得服务，从而改善其延迟性能，由此可能带来的公平性问题将在后面讨论。

算法主要包括初始化、入队和出队 3 个过程。算法维护两个服务链表分别是当前服务链表和下轮服务链表，每次服务时给各个队列分配不超过一个特定大小的令牌数，这样完成一轮服务称为内轮循，然后再开始一轮服务，一直到所有的队列可用的令牌数都用尽为止。具体描述如下：设 AQ[i]是在每轮服务中应给队列 i 分配的令牌数目，Q_{min} 则是所有AQ[i]中最小值，CC[i]表示队列 i 当前可用的令牌数目，SQ[i]记录队列 i 每次内轮循后还剩余的令牌数目，MSQ[i]记录每次内轮循中给队列 i 分配的令牌数目，初始化过程主要找到最小令牌数 $Q_{\min}$ 和置 CC[i]为零。入队过程，当到来的包所对应的队列原来为空时，如果是优先队列则将该队列号加入到当前服务链表，否则加入到下轮服务链表；当队列为满时则丢弃包，否则直接入队即可。出队过程，从当前服务链表头获得服务队列号，每次给该队列分配以内轮循令牌数目 $Q_{\min}$，这样当前可用令牌数目 CC[i]是由前次服务后剩余的可用令牌数和所分配的内轮循令牌数目之和，每次内轮循令牌数目都不大于剩余令牌数 SQ[i]和 $Q_{\min}$。以当前可用令牌数 CC[i]服务队列 i，每次发送一个包就从 CC[i]减去相应的字节数，直到队列为空或没有足够的可用令牌数为止，如果是因没有足够的令牌数而停止服务，且剩余令牌数 SQ[i]和当前可用令牌数 CC[i]的和也不足以服务下一个包，则该队列进入下轮服务队列，否则加入到当前队列尾部。如此直到当前服务链表空为止，结束本轮服务，交换两个服务链表，使得下轮服务链表成为当前服务链表，新的下轮服务链表为空，开始了新一轮的服务。

3 算法的性能分析

3.1 公平性的分析

公平性是衡量调度算法的重要指标，这里采用公平性指数来分析。公平性指数是用两个队列得到服务的归一化业务量之差[2]表示，即如果任何两个队列 i 和 j 在时间$(t_1, t_2]$内持续有数据包等待发送，并且有：

$$[S_i(t_1,t_2)/f_i - S_j(t_1,t_2)/f_j] \le C \tag{1}$$

其中 f_i 定义为 $f_i = Q_i / Q_{\min}$，表示分配给队列 i 的发送速率权重因子。如果 C 是一个与时间间隔$(t_1, t_2]$无关的常数。那么 C 为服务的公平性指数。文献[2]已经证明 NDRR 算法的公平性，新算法，对优先队列采用了特殊服务机制，算法的公平性必然会有所改变，下面将证明只要该队列分配的虚令牌数 Q_C 和漏桶的参数 (σ,ρ) 满足一定的关系，算法的公平性依然能够满足。其中参数 σ 表示漏桶深度，ρ 表示漏牌的到达速率，输出链路的服务速率用 r 表示，Q_i 表示队列 i 分配的令牌数。

引理 1 完成一次轮循的最长服务时间 T_P 满足关系式 $T_P \le \sum Q_i / r + \sigma/(r-\rho)$。

证明 算法中，完成一次轮循服务需要最长时间的情形是：在为多个队列服务，将要结束而进入下一轮循的时候，本来为空的优先队列又有包到达，该队列将加入到当前服务链表，作为本轮任务被立即服务，此后优先队列将单独接受服务直到本轮结束。这样一轮服务时间包括两部分：为多个队列服务的时间 T_t 和单独为优先队列服务的时间 T_C。算法是以 $Q_{\min}$ 为粒度的调度机制，多个队列服务的最大服务量是所有令牌数之和 $\sum Q_i$，这段服务时间可表示为 $T_t = (\sum Q_i)/r$；单独给优先队列的最大服务量是该队列分配的令牌数目 Q_C，由于有漏桶限制，则所能连续服务的最长时间 T_C 应该满足关系 $\sigma + \rho \cdot T_C = r \cdot T_C$ 即 $T_C = \sigma/(r-\rho)$；则一轮循环调度的最长时间周期可以表示如下：

$$T_P \le T_t + T_C = \frac{\sum Q_i}{r} + \frac{\sigma}{r-\rho} \tag{2}$$

这个引理表明即使有优先服务机制的存在，由于漏桶的限制和小粒度服务量的制约，一轮的服务时间仍然是有限的，非优先队列不会无限等待。

引理 2 如果虚令牌数 Q_C 和漏桶的 σ 参数满足关系式:

$$Q_C > Q_{\min}(\sum f_i)\frac{2\rho}{r} \tag{3}$$

和

$$\sigma < \frac{r-\rho}{r}(\sum Q_i) \tag{4}$$

则在任意时间段 $(t_1,t_2]$ 之内该优先服务队列发送的业务量 $S_C(t_1,t_2)$ 满足关系式 $S_C(t_1,t_2) < \sigma + (m+1)\cdot fc \cdot Q_{\min}$，其中 $f_C = Q_C / Q_{\min}$，$m = \left\lceil \frac{t_2 - t_1}{T_p} \right\rceil$，表示 $(t_1,t_2]$ 之内服务的循环轮数目。

证明 设 m 表示从(t_1, t_2)内时段内完成轮循服务的数目，受到前面漏桶的限制，则发送的业务流有 $S_C(t_1,t_2) \le \sigma + \rho(t_2 - t_1) \le \sigma + \rho(m+1)T_p$，根据引理 1 有

$$S_C(t_1,t_2) \le \sigma + (m+1)\frac{\rho}{r}(\sum Q_i) + \frac{\rho}{r-\rho}\sigma(m+1) \tag{5}$$

如果满足式（3）并有 $\sum Q_i = Q_{\min}\sum f_i$，由于虚令牌的分配，则有 $\frac{2\rho}{r} < f_C / \sum f_i$，则式（5）第 2 项变成

$$(m+1)\frac{\rho}{r}(\sum Q_i) < (m+1)\frac{f_C/2}{\sum f_i}(\sum Q_i) = \frac{(m+1)}{2} f_C Q_{\min} \tag{6}$$

对于第 3 项如果满足条件式（4）：$\frac{\sigma}{r-\rho} < \frac{\sum Q_i}{r}$，结合式（3）则有

$$(m+1)\frac{\rho}{r-\rho}\sigma < (m+1)\frac{\rho}{r}\cdot\sum Q_i < \frac{m+1}{2} f_C \cdot Q_{\min} \tag{7}$$

从上面式（6）和式（7）有 $S_C(t_1,t_2)<\sigma+[(m+1)/2]f_CQ_{\min}+[(m+1)/2]f_CQ_{\min}$ 进一步有

$$S_C(t_1,t_2)/f_C<\frac{\sigma}{f_C}+(m+1)Q_{\min} \tag{8}$$

定理　在任意时间间隔 $(t_1,t_2]$ 内，如果虚令牌数 Q_C 和漏桶的 σ 参数满足式（3）和式（4），则 PNDRR 调度算法满足公平性的要求。

证明　对于任意非优先队列 i，在任意时间段 $(t_1,t_2]$ 之内发送的业务量 $S_i(t_1,t_2)$ 依据文献[2]:

$$mQ_{\min}-\mathrm{Max}/f_i\le S_i(t_1,t_2)/f_i \tag{9}$$

其中 m, Max，$Q_{\min}$，f_i 代表的含意和前面各个引理中定义的相同。结合引理 2 的结论式（8）有:

$$S_C(t_1,t_2)/f_C<\sigma/f_C+(m+1)Q_{\min} \tag{10}$$

结合式（9）和式（10）有

$$S_C(t_1,t_2)/f_C-S_i(t_1,t_2)/f_i\le Q_{\min}+\sigma/f_C+\mathrm{Max}/f_i \tag{11}$$

式（11）右边项是一个与时间无关的常数，根据式（1）公平性的定义，可见只要选取的参数满足式（3）和式（4），算法的公平性是能够得到满足的。

3.2 延时特性分析

算法中延迟特性的改善是以增加其它队列的延迟为代价的，但这些增加的延迟时间是分摊到多个队列中，队列越多，分摊到其它各个队列的延迟增量就越少。下面分别分析虚令牌的分配和优先服务机制对延时性能的改善效果。

3.2.1 虚令牌分配对延时性能的改善　根据 Latency-rate server 关于调度算法[6]分析的理论，标准 DDR 算法的延迟上限可用如下公式表示：$D_i\le[(f-f_i)M+(n-1)(m-1)]/r+(m-1)(1/\rho_i-1/r)$，式中 n 表示队列的个数，m 表示队列 i 中包的最大长度，M 表示在所有队列中包的最大长度，r 是输出链路的服务速率，ρ_i 表示为队列 i 分配的服务速率，设 $\rho_{\min}$ 为所有队列中分配的最低速率，定义 $f_i=\rho_i/\rho_{\min}$，则 W_i 表示每个服务队列所对应的权重，$f=\sum f_i$ 表示各个队列所分配权重的和，显然稳定的系统应有 $\sum_{0<i\le n}\rho_i<r$。该式给出了 DRR 算法的延时上限。虚令牌的分配实际上增大了优先服务队列的 f_i,从这个公式中还可以看出增大 f_i 可以减小延迟上界 D_i。

3.2.2　优先服务规则对延时性能的改善　算法中提到的优先服务机制可以让优先队列中的包尽早得到服务，减小包在队列中的等待时间，从而减小包的延迟；同时算法中的虚令牌给优先队列赋予了比实际速率更高比例的令牌数，加上优先队列进入服务链表的机制，优先队列中的包延迟将进一步减小。在实际的系统中包的延迟与业务流的到达特性密切相关，很难精确计算，在这里采用半定量分析方法，考查优先队列中包的延迟上界的变化。

根据引理 1 轮循最长时间周期是 T_p，由两部分组成: $T_P=T_1+T_2$,其中 $T_1=\sum Q_i/r$，$T_2=\sigma/(r-\rho)$,对于优先队列中的包，设在 T_1 时间段内的接受服务的包的延迟上界是 D_1，而在 T_2 时间段内接收服务的包的延迟上界是 D_2。由于包的到达时刻与 T_1 和 T_2 无关，延迟时间上界 D 可表示为 $D=[T_1/(T_1+T_2)]D_1+[T_2/(T_1+T_2)]D_2$　在没有优先服务机制时包延迟上界不小于 D_1，则算法的延迟的改善可表示为 $\Delta D=D-D_1=[T_2/(T_1-T_2)](D_2-D_1)$，由前面的分析可知，在 T_1 时段是以轮循方式为多个队列服务，而 T_2 时段是单独为优先队列服务所得到的延迟上界,必有 $D_2<D_1$，则 $\Delta D<0$，即包的延迟必然减小。减小的延迟量可表示为 $\Delta D=-D_1[T_2/(T_2+T_1)]=-D_1[1/(1+T_1/T_2)]$。由引理 1 的证明有

$$\frac{T_1}{T_2}=\frac{r-\rho}{r}\sum Q_i/\sigma=\frac{r-\rho}{r}Q_{\min}\sum f_i/\sigma \tag{12}$$

从式(12)可以看出，选取较小的 $Q_{\min}$ 可以更有效地减小延迟。不过为了维持调度的复杂度在 $O(1)$,仍然需要保证 $Q_{\min}>\mathrm{Max}$ 这个条件[2]。同时选取较大的漏桶深度 σ，可以使包得到更多缓存，从而让更多的包获得输出链路单独服务的机会，从总体上减小该队列中包的延迟。

3.3 算法复杂度分析

算法中引入虚令牌和漏桶需要额外的硬件，但并没有为调度一个包而引入更多的计算时间。对于服务调度机制的改变，也不会使包调度的计算时间有所变化，因此只要保证 $Q_{\min}>\mathrm{Max}$ 这个条件[2]，调度算法的时间复杂度与 DRR 算法是一样的，仍然是 $O(1)$。

4　仿真实验结果

采用 3 个输入业务源模拟不同业务类队列的输入，3 个输入业务源编号依次是 Source 0～2，包长服从 100 到 1500 byte 之间均匀分布，所有业务源的包到达服从指数分布，其中业务源 Source 0 输入到的是需要改善延迟的低速率低延迟业务对应的优先处理队列，包到达的间隔是平均时间为 3s 的指数分布，它对应队列分配的令牌数在 DRR 和 NDRR 算法中都是 2000 byte，在 PNDRR 中是 6000 byte，其它两个业务源则是服从到达间隔平均时间为 1s 的指数分布,相应的队列在 3 种算法中分配的令牌数都是 6000 byte。输出侧的服务速率都选为 4000 byte/s，3 个队列的存贮器的大小都是 10byte。PNDRR 算法中的漏桶参数 (σ,ρ) 值为(12000 byte，800/3(byte/s))。表 1 是仿真结果，可以看到，在 DRR 算法下，低速率的业务源 Source 0 的延迟较其它两个业务的延迟值更大；而 NDRR 算法中低速率的业务好于其它两个业务，PNDRR 算法使得业务源 Source 0 的延迟降低更多，同时没有明显的增加其它两个业务的延迟时间。

表 1 不同算法的延迟对比

算法	DRR		NDRR		PNDRR	
业务源	Source 0	Others	Source 0	Others	Source 0	Others
延迟（s）	0.31	0.22	0.23	0.25	0.12	0.26

5 结论

提出了具有优先服务机制的 PNDRR 算法。这个算法在 NDRR 算法基础上，采用漏桶与虚令牌分配的机制相结合的策略，对时延敏感业务队列以优先服务策略，有效地降低了业务的调度延迟时间，对低速率业务效果更为明显，同时算法保持了较低的复杂度。通过理论分析证明，选取合适的参数，同样可保证算法的公平性。仿真实验的结果表明性能的改进是显著的。

参 考 文 献

[1] Parekh A K, Gallager R G. A generalized processor sharing approach to flow control in integrated services networks: The single node case. *IEEE/ACM Trans. on Networking*, 1993, 1(3): 344 – 357.

[2] Shreedhar M, Varghese G. Efficient fair queueing using deficit round robin. *IEEE/ACM Trans. on Networking*, 1996, 4(3): 375 – 385.

[3] Macgregor M, Shi W. Deficit for bursty latency-critical flows: DRR++, IEEE International Conference on Networks, Singapore, 2000: 287 – 293.

[4] Kanhere S S, Sethu H. Fair, efficient and low-latency packet scheduling using nested deficit round robin. 2001 IEEE Workshop on High Performance Switching and Routing,, Texas, USA, 2001: 6 – 10.

[5] Mao J M, Moh W M, Wei B. PQWRR scheduling algorithm in supporting of DiffServ. International Communication Confrence, Amsterdam, the Netherlands, 2001: 679 – 684.

[6] Stiliadis D, Varma A. Latency-rate servers: A general model for analysis of traffic scheduling algorithms. Proceedings of IEEE INFOCOM'96, San Francisco, USA, 1996: 111 – 119.

简贵胄： 男，1975 年生，博士生，主要兴趣为通信集成电路设计、宽带通信网络设计.

葛 宁： 男，1970 年生，副教授，主要研究领域为宽带网络、SDH 光网络、通信 ASIC 技术.

冯重熙： 男，1930 年生，教授，主要研究领域通信网络、宽带接入网络、通信设备技术.