PSS：一种提供服务质量保证的区分优先级的分组调度架构

邹君妮 许孙娟 林如俭
(上海大学通信工程学院 上海 200072)

摘 要：为解决融合多媒体应用的互联网的 QoS 问题，该文提出了一种适用于高速分组网络的低复杂度的分组调度通用架构(PSS, Priority-based Scheduling Structure)。PSS 将整个调度分为高、低两个优先级调度过程，在高优先级调度过程，提出了一种带约束条件和速率控制因子的排序优先型算法，避免了带宽抢占现象，有效控制了高优先级业务的服务速率；在低优先级过程，提出了一种改进的帧结构型算法，不仅降低了算法复杂度，减小了硬件实现成本，而且缓解了输出业务流的突发性。最后从数学分析和仿真实验两方面证实了 PSS 架构的可行性和实效性。

关键词：服务质量；分组调度；优先级；排序优先型算法；帧结构型算法
中图分类号：TN915.07 **文献标识码**：A **文章编号**：1009-5896(2007)03-0702-05

PSS: A QoS-Oriented and Priority-Based Packet Scheduling Structure

Zou Jun-ni Xu Sun-juan Lin Ru-jian
(*School of Communication Engineering*, *Shanghai University*, *Shanghai* 200072, *China*)

Abstract: To solve the QoS issue of the Internet including multi-services, a Priority-based Scheduling Structure (PSS) designed for high-speed packet networks is proposed. PSS divides packet scheduling into high-priority section and low-priority section. In the high-priority section, a sorted-priority algorithm with low implementation complexity is presented to avoid bandwidth preemption and to control effectively service rates of high-priority services. In the low-priority section, an improved framed-based algorithm is proposed, which decreases not only the algorithm complexity but also the hardware implementation cost. Computer simulation results as well as theoretic analysis show that the PSS mechanism has excellent performance in terms of the implementation complexity, fairness and delay properties.

Key words: QoS; Packet scheduling; Priority; Sorted-priority algorithm; Frame-based algorithm

1 引言

所谓分组调度就是服务器按照某种调度算法，为到达队列的分组安排输出的先后顺序。根据服务器的内部结构，可以把调度算法分为排序优先型(sorted-priority)和帧结构型(framed-based)两大类。排序优先型算法根据系统势能为到达服务器的每个分组计算一个分组势能，而后按照分组势能大小的特定顺序发送分组。诸如Generalized Processor Sharing(GPS)[1]，Weighted Fair Queueing(WFQ)[1]和VirtualClock[2]等都属于排序优先型算法。排序优先型算法能够为会话的可用带宽和端到端延时提供确切保证，通常具有较好的延迟特性；帧结构型服务器把时间分为固定或可变长度的时间片，允许会话按照特定顺序在属于它的时间片内传输数据。诸如Deficit Round Robin(DRR)[3]和Credit Round Robin(CRR)[4]都属于帧结构型算法。帧结构型算法由于排序操作简单，往往具有较低的算法复杂度。

众所周知，传统的互联网是不提供 QoS(Quality of Service)保证的，路由器采用“一视同仁”的工作方式，“尽力而为”地将所有的业务流送达目的地。这种工作方式对WWW，Ftp，Email 等互联网传统业务是非常合适的，无论是排序优先型调度算法，还是帧结构型调度算法都能很好地满足这些业务的通信要求。随着互联网的网络规模和 IP 业务种类的迅猛发展，互联网正在从当初单纯传送数据向可传送数据、语音和视频的多媒体网络转变。各种实时与多媒体业务的引入，对数据的传输提出了低时延、低抖动等要求。单纯采用排序优先型算法或者帧结构型算法由于不能区分业务等级，不提供延时保证，已经无法支持诸如远程教学、视频点播和网络电视等新应用。为解决互联网所面临的这一困境，本文提出了一种具有服务质量保证的区分优先级的分组调度架构 PSS(Priority-based Scheduling Structure)。它是一种通用架构，可以广泛应用于有分级调度要求的高速分组网络，同时满足实时和非实时业务的服务质量要求。

2 分级调度架构 PSS

总体而言，互联网的业务类型可以分为数据、语音和视频三大类。数据业务指传统的BE(Best-Effort)业务，它属于非实时业务，不需要QoS保证；语音和视频业务属于实时(RT,

2005-06-09 收到，2006-01-20 改回
国家自然科学基金(60377024)和上海市科委科学技术发展基金(04dz12045)资助课题

Real-Time)业务，需要提供QoS保证。在融合实时和非实时业务的网络中，无论是采用帧结构型算法还是排序优先型算法，都无法有效保证实时业务的服务质量，原因如下。(1)帧结构型算法。帧结构型算法的最大优点是算法简单，硬件实现容易。然而它仅仅提供带宽保证，不能有效控制延时和延时抖动，所以无法满足实时业务的QoS需要。(2)排序优先型算法。排序优先型算法能够提供带宽和延时保证，因而非常适合实时业务。不过它是一种公平类算法，根据业务流的平均速率分配带宽，一旦语音、视频和数据业务同时竞争带宽，那么低速率的语音和视频业务实际得到的带宽将远远小于高速率的数据业务得到的带宽，导致语音和视频业务的排队延时过大，服务质量无法保证[5]。其次，互联网上大量传输的仍然是BE业务，如果它们也采用排序优先型算法来调度，硬件成本太高，而且良好的服务质量对数据业务效果不明显。

鉴于上述分析，本文提出了区分优先级的分组调度架构PSS，如图 1 所示。PSS 摒弃了采用统一的调度算法来处理所有业务的传统做法，而是根据业务特征将业务划分为高、低两个优先级，相应地，整个调度过程也被分成高优先级调度和低优先级调度两个调度过程。高优先级调度过程处理的对象是服务优先级高的 RT 业务，采用的是延迟特性好的排序优先型算法；低优先级调度过程处理的对象是服务优先级低的 BE 业务，采用的是复杂度低的帧结构型算法。两个调度过程彼此衔接，两种调度算法相互独立，首先满足 RT 业务的带宽需要，而后将链路剩余带宽分配给 BE 业务。这样既保证了 RT 业务的服务质量，同时又减小了算法的整体复杂度。

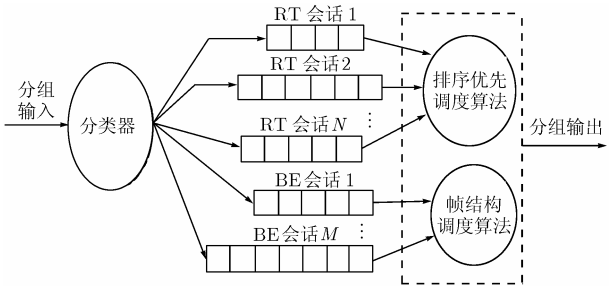

图 1　PSS 调度器的架构

3　高优先级调度过程

首先建立调度模型。假定有 N 个 RT 会话和 M 个 BE 会话共享 C Mb/s 的链路带宽，RT 会话 i 的约定速率(预留带宽)为 r_{hi}，BE 会话 j 的平均速率为 r_{li}。为加以区别，下文开始，所有代表 RT 会话的下标都用 hi 标识，所有代表 BE 会话的下标都用 li 标识。

3.1　高优先级带宽抢占现象

实现 PSS 架构的前提条件是 RT 会话按照约定速率传输数据，而后 BE 会话分享链路剩余带宽。然而，如果将现有的排序优先型算法直接应用于 PSS 的高优先级调度过程，将出现 RT 会话的实际传输速率(以 g_{hi} 表示)远远超出其约定速率 r_{hi} 的情况，本文把这种现象称为高优先级带宽抢占。来看一个具体实例，峰值速率分别为 1Mb/s，1.5Mb/s 和 2.5Mb/s 的 3 个 RT 会话和 5 个 BE 会话共享 10Mb/s 的链路带宽，当采用 WFQ 和 VirtualClock 调度 RT 会话时，RT 会话的实际服务速率 g_{hi} 并没有随 r_{hi} 线性变化，而是最大限度地接近其峰值速率，如图 2 所示。这种带宽抢占现象一方面导致 RT 会话实际得到的带宽大于它申请预留的带宽，出现带宽过剩；另一方面又造成 BE 会话的带宽不足和排队延时过大。究其原因主要有以下两方面。

一是分组提前发送。在输入相同的情况下，同一分组在基于包调度的服务器(以WFQ为代表)中的离开时间落后于它在基于流调度的GPS服务器的离开时间不会超过发送一个最大分组所需的时间[1]。这仅仅给出了包调度落后与流调度的时间上限，事实上，有可能出现在在流系统中尚未开始传输的分组在包系统中被提前发送了。图 3 比较了相同输入时，分组在WFQ服务器和GPS服务器中的离开时间，可以看出，在被服务的 29 个分组中，有 5 个分组在WFQ服务器中的服务时间超前于它在GPS服务器的服务时间。而这种提前发送正是RT会话抢占带宽的一个重要原因。

二是空闲带宽的不合理分配。如果某个RT会话的队列被清空，该会话就从积压状态过渡到空闲状态，系统预留给它的带宽就成为空闲带宽。空闲带宽的分配方式是由排序优先型算法的系统势能的变化率所决定。WFQ的系统势能变化率为 $\frac{\sum_{i=1}^{N} r_{hi}}{\sum_{j\in B(t)} r_{hi}}$，这表明WFQ强制性地将所有的空闲带宽分配给了RT会话，这进一步加剧了带宽抢占现象。

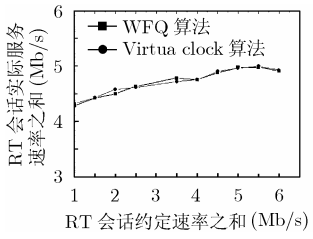

图 2　RT 会话的实际传输速率与约定速率的关系

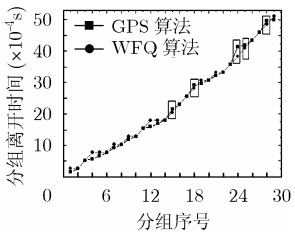

图 3　分组在 GPS 服务器和 WFQ 服务器的离开时间

3.2　PSS-H 算法描述

为了有效控制 RT 会话的传输速率，确保 BE 会话的可用带宽，本文提出了一种改进的排序优先型调度算法，用 PSS-H 来标识，表明该算法适用于 PSS 的高优先级调度过程。

PSS-H 算法定义如下：

(1)在起始时刻，系统势能和所有的会话势能都为零；

(2)系统势能的计算：在系统忙期，系统势能为

$$P(t) = P(t_0) + \frac{1}{\alpha}(t - t_0) \tag{1}$$

(3)分组启动势能和完成势能的计算：设 p_{hi}^{k} 是 RT 会话 i

的第 k 个分组， L_{hi}^k 是 p_{hi}^k 的分组长度， a_{hi}^k 是 p_{hi}^k 到达服务器的实际时刻， S_{hi}^k 是 p_{hi}^k 的启动势能， F_{hi}^k 是 p_{hi}^k 的完成势能，对所有 RT 会话 i 有 $F_{hi}^0=0$，则 p_{hi}^k 的启动势能和完成势能按如下公式计算：

$$\left.\begin{aligned} S_{hi}^k &= \max\{F_{hi}^{k-1}, P(a_{hi}^k)\} \\ F_{hi}^k &= S_{hi}^k + L_{hi}^k / r_{hi} \end{aligned}\right\} \tag{2}$$

(4)数据发送条件：在系统忙期的任一时刻 t，RT 会话 i 的分组能够参与高优先级调度的前提条件是

$$S_{hi}^k \le P(t) \tag{3}$$

(5)数据发送规则：PSS-H 算法优先发送分组完成势能最小的分组。

式(3)给出的约束条件，规定只有那些在流系统中已经或正准备开始传输的分组才能被包系统调度，从而克服了分组在包系统中的提前发送现象。当 PSS-H 服务器服务完一个 RT 分组而需要调度下一个分组时，只能从所有满足式(3)的 RT 分组中选择分组完成势能最小的分组输出。如果当前所有的 RT 分组都不满足数据发送条件，则转入低优先级调度过程，开始传输 BE 分组，直到有 RT 分组满足式(3)，中断低优先级调度过程，返回高优先级调度过程。

式(1)定义的系统势能函数中的 α ($0<\alpha\le 1$)是一个速率控制因子，通过调整 α 可以控制空闲带宽在 RT 会话和 BE 会话之间的分配，进而控制 RT 会话的服务速率。 α 越小，RT 会话占用的空闲带宽越多，服务速率越大。随着 α 的增大，RT 会话的服务速率逐渐减小。当 $\alpha=1$ 时，RT 会话的服务速率就等于其约定速率。所以，可以根据网络的流量特征灵活地调整 α 的大小，以提高空闲带宽的利用率和网络的整体服务质量。当 RT 会话中大量存在的是 CBR 业务时，应该尽可能将空闲带宽留给 BE 会话，因为额外带宽对 CBR 会话而言是多余的，这时 α 的取值应该稍大些；反之，如果 VBR 会话在 RT 会话中占主导，就应当将 α 设小些，为 VBR 会话多分配一些额外带宽，来改善它们的服务质量。例如，假设 VBR 会话的平均速率为 r_{hi}，峰值速率为 p_{hi}，那么它可以按照 r_{hi} 来申请预留带宽，至于 $(p_{hi}-r_{hi})$ 这部分带宽，则可以通过分配空闲带宽来保证。

4 低优先级调度过程

4.1 现有算法的不足

DRR是一种典型的适用于变长分组环境的帧结构型算法，它为每个会话维护一个额度(Quantum)计数器和一个赤字(Deficit)计数器，Quantum表示一轮周期内服务器为会话提供的服务量额度，Deficit表示上一轮周期未用完的额度。轮到发送的会话，只要它的头分组长度小于其剩余额度，就持续为该会话提供服务，服务结束时未用完的额度存储在Deficit计数器中，累积到下一轮周期使用。这种调度方式存在一个问题：约定速率越大的会话其Quantum越大，在一轮周期内连续发送的分组越多，经过服务器输出后会话的突发性越大。同时，Quantum大的会话长时间占用链路，增加了其它会话的排队延时和服务不公平性。

为了减小DRR的服务粒度，提高服务公平性，Do等人提出了CRR算法[4]。CRR按照会话约定速率递减的顺序轮询，即在一轮周期中，速率最大的会话最先接受服务，速率最小的会话最后接受服务。服务器不再持续为一个会话服务至额度用完，而是采用交织方式提供服务，即会话一次只能发送一个分组。一旦所有会话的队头分组长度都超过它们的剩余额度，一轮周期结束。在新一轮周期开始之前，服务器要更新会话的发送额度 CS_i， CS_i 不是一个固定值，而是根据队列状况进行动态调整，它的计算公式为

$$\mathrm{CS}_i = r_i \times \frac{L_*^{\mathrm{HOL}}(\tau)}{r_*(\tau)} + \mathrm{CS}_i \tag{4}$$

其中 τ 是服务器更新发送额度的时刻，$r_*(\tau)$ 代表 τ 时刻积压会话中的最大约定速率， $L_*^{\mathrm{HOL}}(\tau)$ 代表速率为 $r_*(\tau)$ 的会话队头分组的长度。

尽管 CRR 算法在服务粒度和公平性方面具有非常优越的性能，不过将它应用于高速大容量网络仍然存在以下几点不足。

(1)CRR 在发送数据之前需要知道分组长度以判断是否允许发送，这个要求会增加实现的开销。而且对于高速网络而言，获取分组长度的代价往往大于传输分组本身的代价。

(2)服务粒度过小，造成会话多轮周期没有发送机会。CRR 在 DRR 基础上减小了服务粒度，但它更新 CS_i 时，没有控制其最小值，容易出现 CS_i 远远小于会话的头分组长度，会话多轮周期不能发送分组的情况。例如，假设 M 个 BE 会话中，速率最大的会话 S_1 的约定速率是速率最小的会话 S_M 的 10 倍，并且 $L_{l1}=64$ byte， $L_{lM}=1518$ byte，根据式(4)，会话 S_M 要每隔 23 轮周期才能发送一个分组，对于突发业务而言，这种排队延时会很大。

(3)算法复杂度高。为了将会话按照速率大小进行排序，CRR 必须实时跟踪积压会话集合，算法复杂度高达 $O(M)$。

4.2 PSS-L 算法描述

为了让 CRR 算法更好地服务于 PSS 的低优先级调度过程，我们对 CRR 进行了适当改进，改进后的算法用 PSS-L 表示，其特定如下。

(1)不采用速率递减的轮循次序，而是采用 DRR 算法的顺序轮循方式；

(2)系统初始时： $\mathrm{CS}_i=0$；

(3)一轮周期开始之前进行发送额度更新：

$$\mathrm{CS}_i = r_{li} \times \max_{j\in B(\tau)} \left\{ L_{lj}^{\mathrm{HOL}}(\tau) / r_{lj} \right\} + \mathrm{CS}_i \tag{5}$$

其中 τ 代表额度更新时刻， $\max\limits_{j\in B(\tau)} \left\{ L_{lj}^{\mathrm{HOL}}(\tau) / r_{lj} \right\}$ 代表 τ 时刻所有积压的 BE 会话中，头分组长度与会话速率比值的最大值。将 $\max\limits_{j\in B(\tau)} \left\{ L_{lj}^{\mathrm{HOL}}(\tau) / r_{lj} \right\}$ 作为服务粒度是为了尽可能避免会话

在一轮周期中被轮空的现象，以减小输出流量突发性；

(4)当轮循到会话 i 时，服务器不需要知道会话 i 头分组的长度，而是直接发送其队头分组，并在发送过程中得到分组长度，而后从 CS_i 中扣除相应服务量。如果出现 $\mathrm{CS}_i<0$，则将会话序号从链表中删除，透支的服务量将在下一轮周期中扣除。

5　性能分析

衡量分组调度算法性能的指标通常包括复杂度、公平性和延迟特性。公平性一般通过服务公平指数(Service Fair Index, SFI)衡量，延迟特性由最坏公平指数(Worst-case Fair Index, WFI)衡量[6]。由于PSS采用了两种不同的调度机制，所以必须分开讨论它们的性能。

定理1　PSS的算法复杂度是 $O(1)+O(\log M)$。

证明　排序优先型算法的复杂度由更新系统势能所需的代价决定。从PSS-H的系统势能函数定义可以看出，一旦 α 的值给定，PSS-H不需要跟踪积压会话集合，所以PSS-H的复杂度达到排序优先型算法的最低复杂度 $O(1)$；PSS-L服务器进行额度更新时，要对所有会话头分组的 $L_{li}^{\mathrm{HOL}}/r_{li}$ 值进行排序，选择最大值作为下一轮的服务粒度，换言之，一轮周期服务器只要进行一次排序操作，复杂度为 $O(\log M)$，明显比CRR算法的 $O(M)$ 复杂度低。

定理 2　如果从 τ 时刻开始，RT 会话 i，j 一直处于积压期，那么对于从 τ 时刻之后的任意有限时段 $[t_1,t_2)$，会话 i，j 满足

$$\left|\frac{W_{hi}(t_1,t_2)}{r_{hi}}-\frac{W_{hj}(t_1,t_2)}{r_{hj}}\right|\le 2\left(\frac{L_{hi,\max}}{r_{hi}}+\frac{L_{hj,\max}}{r_{hj}}+\frac{L_{h\max}}{C}\right)-\frac{L_{hi,\max}}{C}-\frac{L_{hj,\max}}{C} \tag{6}$$

其中 $W_{hi}(t_1,t_2)$ 代表会话 i 在 $[t_1,t_2)$ 内的服务量，$L_{hi,\max}$ 代表会话 i 的最大分组长度，$L_{h\max}$ 代表所有RT会话的最大分组长度。

上述定理给出了PSS-H算法的SFI，它的证明方法可以参考文献[7]，证明过程略。

定理 3　如果从 t_0 时刻开始，BE 会话 i，j 一直处于积压期，而且在 t_0 时刻之后的有限时段 $[t_1,t_2)$ 内，会话 i，j 经历了 k 轮周期，那么

$$\left|\frac{W_{li}(t_1,t_2)}{r_{li}}-\frac{W_{lj}(t_1,t_2)}{r_{lj}}\right|\le\frac{L_{li,\max}}{r_{li}}+\frac{L_{lj,\max}}{r_{lj}} \tag{7}$$

证明　不失一般性，将 (t_1,t_2) 区间划分为图 4 所示的几个子区间，其中 $[\tau_{n_0+i-1},\tau_{n_0+i})$ 代表第 (n_0+i) 轮周期的持续时间，并且假设会话在第 (n_0+k+1) 轮周期的服务开始时刻大于 t_2。

根据上述假设，有 $W_{li}(\tau_{n_0+k},t_2)=0$。由于会话 i 在第 n_0 轮周期进入积压期，所以它的服务从第 (n_0+1) 轮周期开始，则 $W_{li}(t_1,\tau_{n_0})=0$。

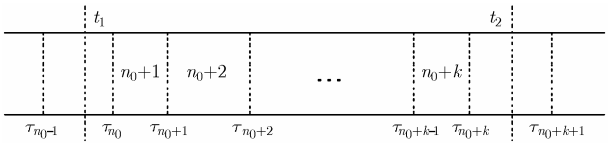

图 4　PSS-L 轮循周期的区间划分

在第 (n_0+i) 轮周期内会话 i 的服务量为

$$W_{li}\left(\tau_{n_0+i-1},\tau^-_{n_0+i}\right)=r_{li}\times\max_{m\in B(\tau_{n_0+i-1})}\left\{\frac{L_{lm}^{\mathrm{HOL}}(\tau_{n_0+n-1})}{r_{lm}}\right\}+\mathrm{CS}_i\left(\tau^-_{n_0+i-1}\right)-\mathrm{CS}_i\left(\tau^-_{n_0+i}\right) \tag{8}$$

在区间 $[\tau_{n_0},\tau_{n_0+k})$ 的 k 轮周期内，会话 i 的总服务量为

$$W_{li}(\tau_{n_0},\tau_{n_0+k})=r_{li}\times\sum_{n=1}^{k}\max_{m\in B(\tau_{n_0+n-1})}\left\{\frac{L_{lm}^{\mathrm{HOL}}(\tau_{n_0+n-1})}{r_{lm}}\right\}+\mathrm{CS}_i(\tau^-_{n_0})-\mathrm{CS}_i(\tau^-_{n_0+k}) \tag{9}$$

所以

$$\begin{aligned}W_{li}(t_1,t_2)&=W_{li}(t_1,\tau_{n_0})+W_{li}(\tau_{n_0},\tau_{n_0+k})+W_{li}(\tau_{n_0+k},t_2)\\&=r_{li}\times\sum_{n=1}^{k}\max_{m\in B(\tau_{n_0+n-1})}\left\{\frac{L_{lm}^{\mathrm{HOL}}(\tau_{n_0+n-1})}{r_{lm}}\right\}+\mathrm{CS}_i(\tau^-_{n_0})-\mathrm{CS}_i(\tau^-_{n_0+k})\end{aligned} \tag{10}$$

根据 PSS-L 的调度规则，在额度更新前的瞬间 CS_i 满足 $-L_{l\max}\le\mathrm{CS}_i\le 0$，即

$$\frac{W_{li}(t_1,t_2)}{r_{li}}\le\sum_{n=1}^{k}\max_{m\in B(\tau_{n_0+n-1})}\left\{\frac{L_{lm}^{\mathrm{HOL}}(\tau_{n_0+n-1})}{r_{lm}}\right\}+\frac{L_{li,\max}}{r_{li}}$$

$$\frac{W_{lj}(t_1,t_2)}{r_{lj}}\ge\sum_{n=1}^{k}\max_{m\in B(\tau_{n_0+n-1})}\left\{\frac{L_{lm}^{\mathrm{HOL}}(\tau_{n_0+n-1})}{r_{lm}}\right\}-\frac{L_{lj,\max}}{r_{lj}}$$

所以

$$\left|\frac{W_{li}(t_1,t_2)}{r_{li}}-\frac{W_{lj}(t_1,t_2)}{r_{lj}}\right|\le\frac{L_{li,\max}}{r_{li}}+\frac{L_{lj,\max}}{r_{lj}}$$

证毕

讨论完 PSS 的复杂度和公平性，最后来分析它的延时抖动。

引理 1　对于 PSS-H 服务器和与之对应的 GPS 服务器，有下式成立

$$Q_{hi,\mathrm{GPS}}(\tau)-Q_{hi,\mathrm{PSS}-H}(\tau)\le\left(1-\frac{r_{hi}}{C}\right)L_{hi,\max} \tag{11}$$

这里 $Q_{hi,\mathrm{GPS}}(\tau)$ 和 $Q_{hi,\mathrm{PSS}-H}(\tau)$ 分别代表 τ 时刻，RT 会话 i 在 GPS 服务器和 PSS-H 服务器中的队列长度。这里只给出结论，它的证明方法可以参考文献[8]。

定理 4　采用 PSS-H 算法调度 RT 会话时，任意 RT 会话 i 满足

$$d^k_{hi,\mathrm{PSS}-H} - a^k_{hi} \le \frac{Q_{i,\mathrm{PSS}-H}(a^k_{hi})}{r_{hi}} + \frac{L_{hi,\max}}{r_{hi}} - \frac{L_{hi,\max}}{C} + \frac{L_{h\max}}{C}$$

证明 根据文献[1]有

$$d^k_{hi,\mathrm{PSS}-H} - d^k_{hi,\mathrm{GPS}} \le \frac{L_{h\max}}{C}$$

其中 $d^k_{hi,\mathrm{GPS}}$ 和 $d^k_{hi,\mathrm{PSS}-H}$ 分别代表 RT 会话 i 的第 k 个分组在 GPS 服务器和 PSS-H 服务器中的离开时间。

根据引理 1，有

$$\begin{aligned} d^k_{hi,\mathrm{PSS}-H} - a^k_{hi} &\le d^k_{hi,\mathrm{GPS}} - a^k_{hi} - \frac{L_{\max}}{C} = \frac{Q_{i,\mathrm{GPS}}(a^k_{hi})}{r_{hi}} + \frac{L_{h\max}}{C} \\ &\le \frac{Q_{hi,\mathrm{PSS}-H}(a^k_{hi}) + (1-\frac{r_{hi}}{C})L_{hi,\max}}{r_{hi}} + \frac{L_{h\max}}{C} \\ &\le \frac{Q_{i,\mathrm{PSS}-H}(a^k_{hi})}{r_{hi}} + \frac{L_{hi,\max}}{r_{hi}} - \frac{L_{hi,\max}}{C} + \frac{L_{h\max}}{C} \end{aligned}$$

证毕

6 仿真结果

本文利用网络仿真软件OPNET建立仿真系统。系统链路传输速率为100Mbit/s，输出队列中一共有20个RT会话和20个BE会话，RT会话的约定速率包括64Kbit/s，1Mbit/s，2Mbit/s和3Mbit/s四种，BE会话的平均速率包括1Mbit/s ，2Mbit/s，4Mbit/s和5Mbit/s 4种。RT会话的业务源采用了泊松源和ON-OFF源两种，BE会话的业务源为ON-OFF源。

图5显示了约定速率为2Mbit/s的RT会话在GPS服务器和ISS-H服务器中的分组离开时间，相较于图3，分组在ISS-H服务器中的离开时间明显落后于在它在GPS服务器中的离开时间，从而有效克服了WFQ算法的分组超前发送现象。图6比较了约定速率为3Mbit/s的RT会话采用WFQ和ISS-H算法调度时的实际服务速率。采用WFQ时，带宽抢占现象明显，会话的服务速率远远大于约定速率，采用ISS-H时，随着 α 的增大，服务速率逐渐减小，当 $\alpha = 1$ 时，服务速率等于约定速率。

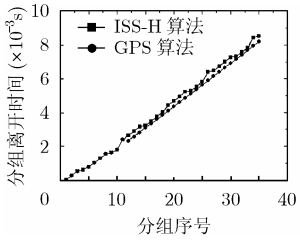

图 5 分组在 ISS-H 服务器中的离开时间

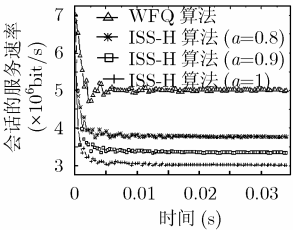

图 6 RT 会话的平均服务速率

图7显示了分组到达平均速率为2Mbit/s的BE会话采用DRR、CRR和ISS-L算法调度时的平均服务速率。DRR算法的服务速率波动性较大，公平性较差。ISS-L算法和CRR算法的服务速率具有相似的平滑特性，不过ISS-L算法的服务速率更接近分组到达速率。图8显示了BE会话采用DRR，CRR和ISS-L算法调度时的平均端到端延时情况，CRR和ISS-L算法因为在服务粒度方面对DRR算法进行了改进，所以有效减小了会话的平均传输延时。

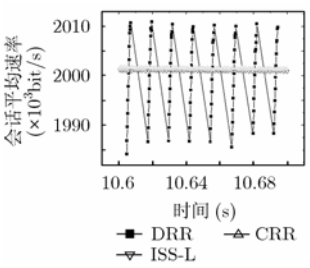

图7 BE会话的平均服务速率

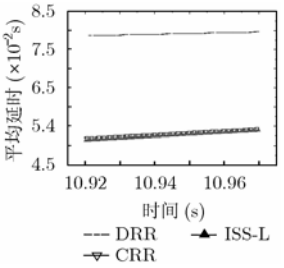

图8 BE会话平均延时

7 结束语

为解决一种调度算法无法保证互联网服务质量的问题，本文提出了一种提供服务质量保证的区分优先级的分组调度架构PSS。根据业务优先级的不同，PSS采用了两个调度过程和两种调度算法。在高优先级过程，通过对排序优先型算法进行重新定义，解决了带宽抢占问题，并实现了对高优先级业务服务速率的灵活控制。在低优先级过程，为降低实现成本，减小业务突发性，提出了一种改进的帧结构型算法。计算机仿真和理论分析表明，PSS架构是复杂度低、公平性和延时特性较好的调度算法，能够广泛应用于高速分组网络，同时满足实时和非实时业务的服务质量要求。

参 考 文 献

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

[2] Zhang L. VirtualClock: A new traffic control algorithm for packet switching networks. *ACM Trans. on Computer Systems*, 1991, 9(2): 101–124.

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

[4] Do V L and Yun K Y. High Performance Switching and Routing. 2003 Workshop on HPSR, California, 24-27 June 2003: 103–110.

[5] Wang Song. Hierarchical Qos Integration for Real-time Systems. Dissertation for the degree of Doctor of philosophy in electrical and computer engineering, University of California, IRVINE, 2003: 30–62.

[6] Bennett J C R and Zhang H. Hierarchical packet fair queueing algorithms. *ACM/IEEE Trans. on Networking*, 1997, 5(5): 675–689.

[7] 杨帆，刘增基. 一种合理共享空闲带宽的分组调度算法. 南京大学学报(自然科学), 2003, 39(2): 246–264.

[8] Bennett J C R and Zhang H. WF²Q: Worst-case fair weighted fair queueing. in Proc. IEEEINFOCOM'96, San Francisco, CA, Mar. 1996: 120–128.

邹君妮： 女，1976 年生，博士生，研究方向为宽带接入、分组交换、光网络.

许孙娟： 女，1981 年生，硕士生，研究方向为数据通信、网络 QoS 性能.

林如俭： 男，1939 年生，教授，研究方向为光通信、宽带接入网.