*i*RGRR/PM：一种新的高速 crossbar 分组调度策略

彭来献[①]　路 欣[②]　田 畅[①]
[①](解放军理工大学通信工程学院　南京　210007)
[②](解放军理工大学第 63 研究所　南京　210007)

摘　要：*i*RGRR(iterative Request-Grant-based Round-Robin)算法是一种输入排队 crossbar 调度算法，具有硬件易实现、可扩展性强、性能优良等优点。在此基础上，该文提出一种新的高速 crossbar 分组调度策略 *i*RGRR/PM(*i*RGRR with Packet Mode)，可以更好地支持 IP 分组的调度，能够被应用于高速、大容量的路由器中。与 *i*RGRR 算法相比，*i*RGRR/PM 简化了分组输出重组模块的设计，并且提高了 crossbar 的带宽资源利用率。文中简单分析了两种算法间的分组时延关系，并进行了详尽的仿真研究。结果表明：在相同条件下，*i*RGRR/PM 算法具有更高的吞吐量，尤其在非均匀业务流下能获得接近 100%的吞吐量；调度长分组时，*i*RGRR/PM 算法具有更好的时延性能。

关键词：路由器；输入排队；分组调度算法；*i*RGRR/PM
中图分类号：TN915.07　　**文献标识码**：A　　**文章编号**：1009-5896(2007)07-1612-05

*i*RGRR/PM: A New Packet Scheduling Scheme for High-Speed Crossbars

Peng Lai-xian[①]　Lu Xin[②]　Tian Chang[①]
[①](*Institute of Communications Engineering*, *PLA Univ. of Sci. & Tech.*, *Nanjing* 210007, *China*)
[②](63 *Research Institute*, *PLA Univ. of Sci. & Tech.*, *Nanjing* 210007, *China*)

Abstract: *i*RGRR(iterative Request-Grant-based Round-Robin) is a scheduling algorithm for input-queued crossbars, which has many good features, such as simple, scalability and fine performance. This paper proposes a new packet scheduling scheme based on *i*RGRR, called *i*RGRR/PM (*i*RGRR with Packet Mode), for high-speed crossbars. *i*RGRR/PM algorithm is appropriate to schedule IP packet, and can be used in routers with high-speed and large capacity. Compared to *i*RGRR, *i*RGRR/PM not only simplifies the design of packet output reassembly module, but also improves the bandwidth utilization of crossbar. The relation of packet delay between two algorithms is briefly analyzed, and simulation studies is done in detail. The results show that *i*RGRR/PM achieves higher throughput under the same circumstances, especially, reaches 100% throughput under nonuniform traffics. In addition, *i*RGRR/PM provides better performance of delay for larger packets.

Key words: Router; Input-queued; Packet scheduling algorithm; iterative Request-Grant-based Round-Robin with Packet Mode (*i*RGRR/PM)

1　引言

输入排队(Input-Queued，IQ)crossbar 作为一种简单、高效的高速交换结构，近年来被广泛应用于高速路由器中[1,2]。在这种交换结构中，输入队列一般采用虚拟输出排队技术(Virtual Output Queueing，VOQ)，即一个输入端为每一个输出端维护一个FIFO(First In First Out)队列。这是为了避免单一FIFO排队带来的队头阻塞问题。另外，为了提高传输效率和简化控制，crossbar一般采用定长交换技术，IP 分组在交换前先划分成固定长度的“信元”(通常为 64 byte 大小)，经crossbar传输在输出端重组后再发送到链路上去。一个$N\times N$规模的基于IP分组的IQ crossbar逻辑结构如图 1 所示。

图 1 中虚线框内的组成部分只处理信元。IP 分组从输入端进入路由器，经过查表、报头处理后到达输入端分割模块(Input Segmentation Module，ISM)，ISM 将 IP 分组分割成固定长度的信元，然后将信元送入到正确的 VOQ 中。调度算法每个“时隙”(传输 1 个信元的时间间隔)执行一次，解决信元输入/输出竞争，控制 crossbar 工作，保证信元无

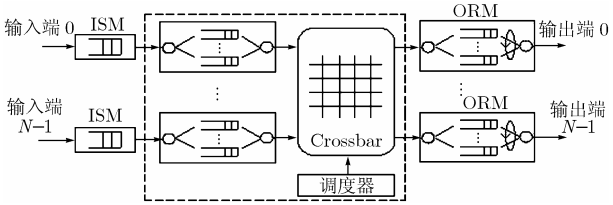


图 1　基于 IP 分组的 IQ crossbar 逻辑结构

2005-12-01 收到，2006-07-03 改回

冲突地传输。经过调度，信元到达输出端后必须经过输出端重组模块(Output Reassembly Module，ORM)，只有当属于同一分组的所有信元全部到达后，经过重组才能被发送到输出链路上。目前，众多研究成果都是研究调度信元时的吞吐量、时延等性能[3–6]，很少关注调度分组时的性能。然而，在路由器中，分组吞吐量、时延等指标更能衡量路由器的性能。文献[7–9]首先提出了分组模式调度概念，主要目的是为了简化输出端重组模块的设计，提高crossbar的带宽利用率和分组调度性能。分组模式调度是指对调度算法增加一条规则：一旦属于某个分组的第 1 个信元开始被调度并转发到输出端，则该输入/输出端连接一直保持到此分组所有的信元传送完毕，期间其它分组的信元不能被转发到该输出端。

文献[3]提出一种 IQ crossbar 调度算法 *i*RGRR(iterative Request-Grant-based Round-Robin)，克服了同类算法的可扩展性差的问题，并且具有硬件易实现、性能优良等优点，能够支持 64～128 个 10Gbps 端口，可被应用于高速、大容量的交换结构中。本文在 *i*RGRR 算法基础上提出一种新的支持分组模式的调度策略——*i*RGRR/PM(*i*RGRR with Packet Mode)，能够更好地支持 IP 分组的调度，获得优良的分组吞吐量、时延等性能。

2　相关背景

2.1　信元模式调度

在图 1 中，一个IP分组被分割成多个信元，在这种信元模式调度下，属于不同分组的信元交叉重叠到达输出端，并且当属于同一个分组的所有信元到达之前，先到达的信元只能缓存在ORM中。因此每个输出端需要N个逻辑(或物理)独立的ORM，分别缓存来自N个输入端的分组，整个交换结构总共需要N^2个。如果多个ORM同时发送，那么还存在分组发送竞争，特别当短分组大量出现时，这种现象更加突出，会带来额外的分组时延。另外，在信元中需要增加属于那个分组的标识信息，以便放入正确的ORM。

很显然，在分组模式调度下，属于同一个分组的信元连续、不间断地被发送到输出端，一个 ORM 就能够完成输出重组的任务，不仅减少了 ORM 的数目，而且不存在 ORM 发送冲突，大大简化了 ORM 的设计，此外，信元不需要标识分组的信息，减少了数据额外负荷，也就提高了 crossbar 的带宽利用率。

2.2　*i*RGRR 算法

*i*RGRR算法的基本思想建立在著名的*i*SLIP算法[5](该算法被应用于CISCO12000 系列高速路由器[3])基础之上，但是通过简化执行步骤和减少调度开销，克服了同类算法可扩展性差的缺陷，可以支持更多、速率更高的端口。另外，*i*RGRR算法调度器具有 2 个$N\times N$规模向一个 $2N\times 2N$规模的扩展能力，这也是同类算法不具备的[3]。在*i*RGRR算法中，每个输入/输出端都有一个round-robin仲裁器，用于解决信元竞争。*i*RGRR算法采用多次迭代的策略提高调度效率，每次迭代按照“请求-许可”两个步骤依次执行：

步骤 1　请求(request)　如果一个未匹配的输入端有信元等待发送，根据 round-robin 规则，输入端仲裁器选择某个 VOQ 的请求，然后将此请求发送给相应的输出端仲裁器。(解决输出端竞争)

步骤 2　许可(grant)　如果一个未匹配的输出端接收到多个请求，输出端仲裁器根据 round-robin 规则从中选择一个请求，并向发出该请求的输入端发送许可信号，这样就建立了一个连接。(解决输入端竞争)

3　*i*RGRR/PM 分组调度策略

为了遵循分组模式调度的规则，在 *i*RGRR/PM 算法中，那些在上一个时隙中传送了分组但尚未传送完毕的输入/输出端连接，必须在下一个时隙中继续保持。除此之外，*i*RGRR/PM 基本上与 *i*RGRR 算法一致，每个时隙包含多次迭代过程，在每一次迭代中，同样分为“请求”和“许可”两个步骤。

步骤 1　请求　如果输入端在前一个时隙内建立了连接并且该连接在本时隙继续保持，则仲裁器向该连接的输出端发送请求；否则仲裁器发送请求的方法与 *i*RGRR 算法相同。

步骤 2　许可　同 *i*RGRR 算法。

为了提高调度效率，避免输入端向已经匹配的输出端发送请求，在 *i*RGRR/PM 算法的“请求”步骤中，输入端仲裁器选择某个请求时要保证其对应的输出端尚未匹配。本文使用一种简单的“输出端忙闲通知机制”解决这个问题，基于如下事实：在一个时隙内，如果在某次迭代中一个输入端仲裁器向某个空闲的输出端发出请求，则该输出端此次迭代后一定匹配，直到本时隙结束。每个输入端仲裁器与 N 条控制线相连，分别对应一个输出端的状态。状态线上只要“HIGH”和“LOW”两种状态，分别表示“忙”和“闲”。这样，根据状态线指示的状态，输入端仲裁器可以屏蔽掉那些发送到“忙”的输出端的请求，从而避免了向已经匹配的输出端发送无用的请求。这种机制只稍微增加输入端仲裁器实现的空间复杂性，并不需要交互其它额外调度开销。

在 *i*RGRR/PM 算法具体实现时，只需要对 *i*RGRR 算法中输入端仲裁器做简单的改动。输入端仲裁器增加一个“保持连接”控制信号 KC(1 bit)和一个寄存器 LRS(log$N$ bit)。LRS 记录最近一次连接选择的 VOQ 号。当一个分组在输入/输出端连接上开始发送时，KC 设置为’1’，当此分组发送完毕后设置为’0’。在调度时，若仲裁器 KC 为’1’，则将 LRS 的值作为请求发出；否则按照与 *i*RGRR 算法中输入端仲裁器相同的方式选择和发送请求，并且更新 LRS 的值。当分组在一个连接上传送期间，状态线一直标识该连接的输出端为“忙”，其它输入端仲裁器也就不会向此输出端发出请求。只有当 KC 从‘1’变为‘0’时，才将相应的状态线设

置为闲状态。实际上，如果将 KC 和状态线在每个时隙结束时强制设为‘0’和闲，那么 *i*RGRR/PM 算法的执行结果就是 *i*RGRR 算法在信元模式下的执行结果。因此，仲裁器根据不同的调度模式决定 KC 和状态线如何进行设置，从而完成信元模式或分组模式的调度任务。

4 分组时延分析

定义 1 分组时延 整个分组在交换结构内停留时间(单位：时隙)，是指分组内最后一个信元到达 ISM 的时刻与该信元输出到链路上的时刻之间的时间间隔。

*i*RGRR 和 *i*RGRR/PM 算法的分组时延由 VOQ 等待时间和 ORM 重组时间组成，一个长度为 L 的分组进入交换结构后即使立即被调度、重组输出，其最小分组时延也为 $2(L–1)$ 个时隙。本节通过建立简化的队列模型分析两种算法间的分组时延关系。

只考虑某一个输出端和所有到达该输出端的分组，这些分组分布在不同的输入端 VOQ 中。为了用一个简单的队列模型来估计 VOQ 等待时间，忽略输入端竞争，交换结构中只要有到达输出端的分组，输出端总不会空闲。对于 *i*RGRR 和 *i*RGRR/PM 这类算法，当负载很小或者较大时，这种假设更加符合实际。此外，为了进一步简化队列模型，假设分组到达过程服从泊松分布。

在上述假设前提下，此队列模型是一个M/G/1 排队系统，分组平均到达速率为 λ^P (分组/时隙)，平均分组长度为 $1/\mu^P$ (信元/分组)，队列服务容量C为 1(信元/时隙)，平均服务时间 $E[S]=1/(\mu^P\cdot C)=1/\mu^P$ (分组/时隙)。两种算法调度的差别反映在不同的队列服务规则。*i*RGRR算法可以同时为多个到达同一输出端的分组服务，相当于处理器共享(processor sharing)或者是round-robin服务；在*i*RGRR/PM算法调度下，到达同一输出端的分组只能一个接一个的串行传送，相当于FIFO服务[8,9]。根据排队论中的结论[10]，*i*RGRR 和 *i*RGRR/PM 算法的平均VOQ等待时间 $E\left[D_1^{\mathrm{VOQ}}\right]$，$E\left[D_2^{\mathrm{VOQ}}\right]$ 分别为

$$E\left[D_1^{\mathrm{VOQ}}\right]=\frac{\rho E[S]}{1-\rho} \tag{1}$$

$$E\left[D_2^{\mathrm{VOQ}}\right]=\frac{\rho E[S]}{1-\rho}\times\frac{1+C_b^2}{2} \tag{2}$$

其中 $\rho=\lambda^P\cdot E[S]=\lambda^P/\mu^P$ 为利用因子(或负载)，C_b 为服务时间的方差系数，这里等价于分组长度的方差系数，即 $\frac{\sqrt{D[L]}}{E[L]}$。

假设 D^{ORM} 表示重组时间，在信元模式下，ORM重组时间上限为 $N\cdot L_{\max}$ 个时隙[11]。很显然，对于 *i*RGRR算法，$L_{\min}\le E\left[D_1^{\mathrm{ORM}}\right]\le N\cdot L_{\max}$。在分组模式下，分组串行到达ORM，分组平均重组时间 $E\left[D_2^{\mathrm{ORM}}\right]$ 就等于分组平均长度 $E[L]$。根据式(1)和式(2)，*i*RGRR和*i*RGRR/PM算法的平均分组时延分别为

$$E[D_1]=\frac{\rho E[S]}{1-\rho}+E\left[D_1^{\mathrm{ORM}}\right],\quad L_{\min}\le E\left[D_1^{\mathrm{ORM}}\right]\le N\cdot L_{\max} \tag{3}$$

$$E[D_2]=\frac{\rho E[S]}{1-\rho}\times\frac{1+C_b^2}{2}+E[L] \tag{4}$$

当负载较大时，有 $\frac{\rho E[S]}{1-\rho}\gg E\left[D_1^{\mathrm{ORM}}\right]$ 和 $\frac{\rho E[S]}{1-\rho}\gg E[L]$，所以在这种情况下

$$E[D_1]\approx E\left[D_1^{\mathrm{VOQ}}\right]=\frac{\rho E[S]}{1-\rho} \tag{5}$$

$$E[D_2]\approx E\left[D_2^{\mathrm{VOQ}}\right]=\frac{\rho E[S]}{1-\rho}\times\frac{1+C_b^2}{2} \tag{6}$$

根据式(5)和式(6)，引入分组模式增益[8,9]的概念，定义如下：

定义 2 分组模式增益 *i*RGRR 算法的平均分组时延与 *i*RGRR/PM 算法的平均时延的比值，用 G 表示，那么 $G=\frac{2}{1+C_b^2}$。

根据上述模型和假设，如果分组长度分布具有较小的 $C_b(<1)$，则*i*RGRR/PM算法的平均分组时延小于*i*RGRR算法的；如果分组长度分布具有较大的$C_b(>1)$，则*i*RGRR算法具有较小的平均分组时延。

5 仿真结果与性能分析

像*i*RGRR/PM这类实用的启发式调度算法，解析分析十分困难[5,12]，因此，计算机仿真是一种有效而广泛采用的研究手段。本节主要对基于IP分组的交换结构中*i*RGRR和*i*RGRR/PM算法的平均分组时延性能以及它们之间的关系进行仿真研究，验证第 4 节中的分析结果，并分析两种算法的最大吞吐量性能。仿真模型采用一个 16×16 的IQ crossbar，仿真长度为 1,000,000 个时隙。两种算法的迭代次数均为 4。

5.1 性能参考模型

众所周知，在相同流量到达下，输出排队(Output-Queued，OQ)crossbar 具有最优的性能，因此本文将此作为性能参考模型。在 OQ crossbar 中，所有信元只在输出端进行缓存，有两种排队方法：(1)FIFO-OQ：每个输出端有一个 FIFO，信元先到达该 FIFO，再分发到各个合适的 ORM；(2)VIQ-OQ：一个输出端为每个输入端维护一个 FIFO，类似于 VOQ，这种排队技术被称为 VIQ(Virtual Input Queueing)，VIQ 同时具有 ORM 的功能。前者要求存储器的工作速率是所有输入端速率之和，后者只要求与单个输入端速率相等即可。然而两者都要求 crossbar 工作速率是所有输入端速率之和，由于受到 crossbar 速率的限制，OQ

crossbar 虽然性能优良，但是不适于高速、大容量的处理场合。在仿真研究中，我们同时考虑这两种 OQ crossbar。

5.2 业务流量模型

在交换结构中，分组经过分割到达输入队列或crossbar，相当于信元突发到达。因此，分组到达可用信元突发过程代替，每次突发表示一个分组到达。假设 λ_i^P，λ_{ij}^P 分别表示输入端i和VOQ_{ij}的分组平均到达速率(即负载)。分组到达过程使用两状态的ON-OFF模型模拟，输入端处于ON状态表示一个分组正在接收中，OFF状态表示没有分组到达。分组长度，即包含的信元数，等于ON状态持续的时隙数，用离散随机变量 Φ 表示。业务流量模型取决于服从不同分布的 Φ，本文考虑如下两种业务流量模型：

(1)均匀业务流模型　在这种流量模型中，$\lambda_i^P=\lambda^P$, $\lambda_{ij}^P=\lambda^P/N$ 并且 $\lambda_i=\lambda,\lambda_{ij}=\lambda/N$，其中 $0\le i, j\le N-1$。分组长度 Φ 服从[1, 160]的均匀分布，这是由于IP分组长度受到MTU的限制，一般不超过 10kbyte，例如在ATM的AAL5 中规定MTU为 9180 byte[13]。如果一个信元为64byte，分组最多包含 160 个信元。均匀业务流量中分组长度的方差系数C_b≈0.584。

(2)非均匀业务流模型　在这种流量模型中，$\lambda_i^P=\lambda^P$, $\lambda_{ij}^P=\lambda^P/N$，$\lambda_i=\lambda$, $\lambda_{ii}=80\lambda/(N+79)$, $\lambda_{ij}=\lambda/(N+79)$, $(j\ne i)$，$0\le i,j\le N-1$。在输入端 i 中，目的端为 i 的分组都是长分组，长度为 80；否则为短分组，长度为 1，即

$$\Phi=\begin{cases}80, & i=j\\ 1, & i\ne j\end{cases} \tag{7}$$

非均匀业务流量中分组长度的方差系数C_b≈1.204。采用这种流量是为了分析长、短分组时延性能的相互影响。

5.3 均匀业务流

在相同的均匀业务流下，图 2 比较了*i*RGRR，*i*RGRR/PM算法和OQ crossbar的平均分组时延。在任何负载下，*i*RGRR/PM算法的平均分组时延比*i*RGRR算法的小。当负载较小时($\lambda<0.6$)时，*i*RGRR/PM算法的平均分组时延性能甚至比FIFO-OQ的优良。按照前面的理论分析，说明对于方差系数较小(≈0.584)的分组长度，*i*RGRR/PM算法获得较大的分组模式增益，使得分组时延相对更小。在均匀业务流下，仿真结果与理论分析结果相吻合。在VIQ-OQ中，分组一旦进入交换结构立即被放入ORM中重组，等待输出，效率最高，因此VIQ-OQ中所有分组的平均时延性能最优，如图 2 所示。另外从图中可以看出，在均匀业务流下，*i*RGRR和*i*RGRR/PM算法的最大吞吐量[1)] 都接近 100%。

为了分析分组长度对分组时延的影响，分别考虑 3 种长度服从[1, 16]，[1, 32]和[1, 64]均匀分布的分组，平均分组时延的比较如图 3 所示。在相同的分组到达下，四者的关系与图 2 中的特征一致；对于同一个算法，平均分组时延与平均分组长度成正比。

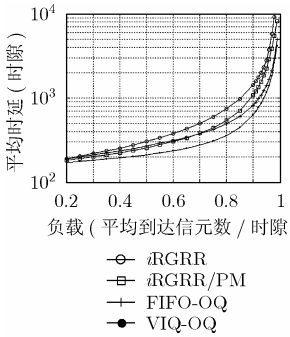


图 2　均匀业务流下的平均分组时延

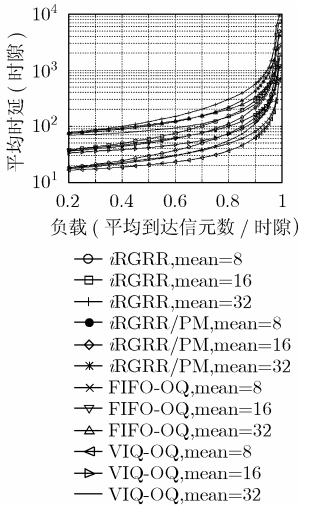
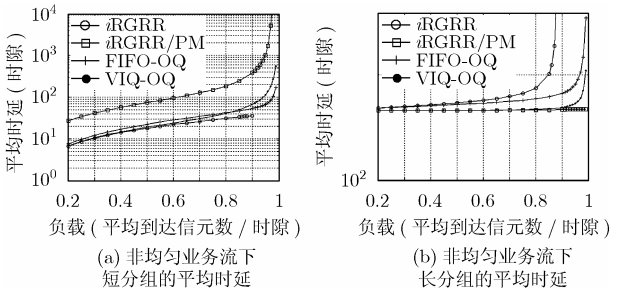


图 3　均匀业务流下分组长度对分组时延的影响

5.4 非均匀业务流

在非均匀业务流下，主要分析长分组与短分组时延性能的相互影响。图 4(a)，4(b)分别是短分组(1 个信元长度)、长分组(80 个信元长度)的平均分组时延。从中发现两个有趣的现象。(1)*i*RGRR 算法最大吞吐量虽然在 88%左右，但它的短分组时延性能最好，*i*RGRR/PM 算法的性能最差；对于长分组，结论相反。(2)对于短分组，FIFO-OQ 在 $\lambda>0.85$ 时获得比 VIQ-OQ 更好的分组时延性能。



(a) 非均匀业务流下短分组的平均时延　　(b) 非均匀业务流下长分组的平均时延

图 4

产生上述现象的主要原因有：(1)*i*RGRR 算法公平对待信元，因此在调度时短分组可以被更快地传送到输出端，分组时延较小。但这是以长分组时延性能性能下降为代价得到的，因为短分组的频繁传送可能使长分组长时间被阻塞在输入队列中，特别是在高负载的情况下，这种现象致使队列系统不稳定，造成系统最大吞吐量在 88%左右。相对而言，*i*RGRR/PM 算法公平对待分组，同时也不关心分组的大小，调度时长分组会长期占有某个连接，会对长分组更有利，当 $\lambda>0.9$ 时，这种优势得到更加显著地体现。虽然短分组性能较差，但保证了系统最大吞吐量基本达到 100%。这也说明在非均匀业务流下，*i*RGRR/PM 算法能改善吞吐量性能。(2)在 FIFO-OQ 中，当负载较高时，短分组频繁到达，由于 FIFO 每个时隙只能向 ORM 输出一个信元，因此在输出端

1)最大吞吐量指队列系统从稳定状态向不稳定状态转换的临界点时的负载(×100%)

短分组可能阻塞长分组的重组，短分组能更快地重组输出，使得比 VIQ-OQ 具有更好的短分组时延性能。而在 VIQ-OQ 中，短分组无法影响长分组的重组，并具有相同的概率被发送到输出链路上，所以不能像 FIFO-OQ 那样以牺牲长分组时延性能来换取更好的短分组时延性能。故而出现上述现象。(3)所有分组在 VIQ-OQ 中被公平对待，而 *i*RGRR/PM 算法在一定程度上更加有利于长分组的调度。所以，在这种非均匀业务流下，*i*RGRR/PM 算法甚至比 VIQ-OQ 具有更小的长分组时延。

总而言之，除了在 VIQ-OQ 中长分组和短分组时延性能影响不大之外，在其它情况下相互影响明显，在 *i*RGRR 算法和 FIFO-OQ 中，调度时对于短分组更有利，在 *i*RGRR/PM 算法中，对于长分组更有利。

6 结束语

本文提出了一种新的分组模式的调度策略 *i*RGRR/PM，它继承了 *i*RGRR 算法良好的可扩展性，同时简化了输出重组模块 ORM 的设计，并且提高了 crossbar 的带宽资源利用率。文中分析了两种算法间的分组时延关系，并进行了详尽的仿真验证。结果表明：在相同条件下，*i*RGRR/PM 算法具有更高的吞吐量，尤其在非均匀业务流下能获得接近 100% 的吞吐量。分组时延与分组长度的方差系数有关，当方差系数小于 1 时，*i*RGRR/PM 算法的平均分组时延较小，在性能和实现上都比 *i*RGRR 算法具有优势；当方差系数大于 1 时，*i*RGRR/PM 算法的平均分组时延较大，但是有利于长分组的调度。从硬件实现复杂性，crossbar 的带宽利用率和综合性能方面考虑，*i*RGRR/PM 算法能够更好地支持 IP 分组的调度，可被应用于太比特(Tbps)路由器中。

参 考 文 献

[1] Partridge C, Carvey P, and Burgess E, *et al*.. A 50Gb/s IP router[J]. *IEEE/ACM Trans. on Networking*, 1998, 6(3): 237–248.

[2] Cisco Inc. Cisco 12000 series——Internet Router. Product Overview[EB/OL], http://www.cisco.com, Oct. 2001.

[3] Peng L X, Tian C, and Zheng S R. *i*RGRR: A fast scheduling scheme with little control messages for scalable crossbar switches[A]. Proceedings of 7th IEEE High Speed Networks and Multimedia Communications, Toulouse, France[C], 2004: 191–202.

[4] Anderson T, Owicki S, and Saxe J, *et al*.. High speed switch scheduling for local area networks[J]. *ACM Trans. on Computer Systems*, 1993, 11: 319–352.

[5] Mckeown N. Scheduling algorithm for input-queued cell switches[D]. [Ph.D. Thesis], USA: UC Berkeley, 1995.

[6] 彭来献, 田畅, 郑少仁. 高速 crossbar 控制算法 *i*DRR 及其性能分析[J]. 电子学报, 2003, 31(10): 1465–1468.

[7] Marsan M A, Bianco A, Giaccone P, Leonardi E, and Neri F. Scheduling in input-queued cell-based packet switches[A]. IEEE Globecom 99, Rio de Janero, Brasil[C], December 1999: 1227–1235.

[8] Marsan M A, Bianco A, Giaccone P, Leonardi E, and Neri F. Packet scheduling in input queued cell-based switches[A]. IEEE Infocom. 2001, Anchorage, Alaska[C], April 2001: 1085–1094.

[9] Marsan M A, Bianco A, Giaccone P, Leonardi E, and Neri F. Packet-mode scheduling in input-queued cell-based switches[J]. *IEEE/ACM Transactions on Networking*, 2002, 10(5): 666–678.

[10] 苏兆龙. 排队论基础[M]. 成都: 成都科技大学出版社, 1998.5, 第 1–3 章.

[11] 孙志刚, 卢锡城. 路由器 IP 报文的重组和调度[J]. 计算机工程, 2002, 28(5): 38–40.

[12] 彭来献. 高速路由器交换体系与调度算法的研究[D]. [博士论文], 南京: 解放军理工大学, 2004, 5.

[13] 谢希仁. 计算机网络(第四版)[M]. 北京: 电子工业出版社, 2003: 186–187.

彭来献： 男，1978 年生，讲师，博士，研究方向为高速交换体系及其调度算法、Ad hoc 网络 MAC 层协议.

路 欣： 女，1979 年生，工程师，硕士，主要从事 CORBA 技术在仪器仪表中的应用研究.

田 畅： 男，1963 年生，副教授，博士，中国电子学会高级会员. 主要从事宽带交换技术、网络安全和无线分组网的研究. 在国内外有关刊物、会议发表论文 40 余篇.