一种基于高速弹性分组环的线性逼近公平算法

柳立峰, 张 雷, 程时端
(北京邮电大学网络与交换国家重点实验室, 北京 100876)

摘 要: 弹性分组环(RPR)中最关键的技术之一是环路分布式公平算法. IEEE802.17 工作组制定的 RPR 草案中关于公平算法存在一些待完善的问题, 比如在高速的网络中存在较长的收敛时间, 同时对于非平衡流带来的永久性震荡现象也无法消除. 本文针对上述这些问题提出了一种新的公平算法. 仿真结果表明算法不仅能够消除非平衡流问题, 而且能够公平地控制站点之间带宽的分配.

关键词: 分布式公平算法; 弹性分组环; 非平衡流; 虚拟目的地队列
中图分类号: TN915 文献标识码: A 文章编号: 0372-2112 (2005) 01-0001-05

A Linear-Approach Fairness Algorithm for High-Speed Resilient Packet Ring

LIU Li-feng, ZHANG Lei, CHEN Shi-duan
(State Key Laboratory of Networking and Switching, Beijing University of Post & Telecommunication, Beijing 100876, China)

Abstract: One of the key techniques of Resilient Packet Ring (RPR) is ring distributed fairness algorithm. While some deficiencies exist in the fairness algorithm recommended by current draft standard of RPR (IEEE802.17 draft), such as long convergence time of fairness algorithm in high-speed RPR network and serious throughput oscillation caused by unbalanced traffic flow. This paper proposes a new fairness algorithm to solve these problems. The simulation results show that this algorithm not only solves the unbalanced traffic problem but also controls fairly the bandwidth allocation among stations.

Key words: distributed fairness algorithm; resilient packet ring; unbalanced traffic flow; virtual destination queue

1 引言

RPR(Ressilient Packet Ring)是一种基于包交换的城域网技术. 它的拓扑类似于传统的 FDDI[1], Token Ring[2] 网络采用双环连接的结构, 但环上站点的数据发送不需要由令牌来决定, 不同站点之间的数据传输允许空间重用, 从而提高了带宽的利用率[3]. RPR 相对于另外一种常用的城域网技术 SDH 的优势在于它在资源分配上更加灵活并同时具备 SDH 的快速保护倒换能力(< 50ms). 图 1 给出了 RPR 的结构示意图.

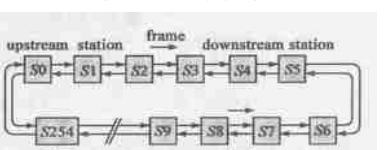

图 1 RPR 结构示意图

RPR 网络 MAC 层的核心功能之一是环网公平控制算法. 即当网络发生拥塞时, 能够控制环上站点在拥塞链路上分配到公平的带宽. 目前城域网带宽增长十分迅速, RPR 已经发展到支持 10Gbps 或更高速率, 在高速 RPR 网络中, 公平算法的收敛时间和算法的复杂度成为至关重要的因素. 而且在高速 RPR 网中, 如果拥塞发生时出现非平衡流[4], 则有可能导致比较严重的流量振荡. 所以我们有必要设计一种在高速和低速 RPR 网络中普遍适用的公平算法.

RPR 草案[3](IEEE802.17 draft)中采用了基于反馈拥塞机制的公平控制算法: 当拥塞发生时, 将拥塞站点(检测到拥塞发生的站点被称为拥塞站点)的本地发送速率作为反馈控制速率广播给上游站点, 上游站点用接收到的反馈控制速率来限制在环上的注入速率, (在本文中, 我们将上游站点在环上的注入速率的阀值称为本地限速速率). 由于上游站点接收到的反馈速率均相等, 保证了它们向环上注入相同的流量, 从而实现了带宽分配的公平性. 但此算法存在一些未解决的问题: 首先, 当环上存在非平衡流时, 此算法会导致永久性的流量振荡; 其次由于控制包在环路上的周期广播和链路的传播延时, 此算法会经历较长的收敛时间; 同时单发送队列的设置还会造成 HOL 问题[3] 等. 为了解决这些问题, 有很多研究工作对该算法进行了改进. 其中文献[5]中提出了 DBRR 算法, 基本思想是拥塞站点对经过的所有会聚流进行基于 GPS[6] 的虚时计算, 虚时的计算是为了准确得到每个会聚流的公平速率. DBRR 在理论上是一种很好的公平算法, 但由于它是通过定期对经过拥塞站点的所有会聚流统计字节计数来估算会聚流的到达速率和虚时, 这种估算存在较大的误差, 同时也带来了额外的处理开销. 文献[4]提出了 MC-SRP 算法, 其基本思想

收稿日期: 2003-12-17; 修回日期: 2004-09-01
基金项目: 高校博士点基金/宽带因特网资源管理和业务控制技术0项目(No. 20010013003)

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

是在每个站点构造接入树以及拥塞链路相关的树干集合, 同时拥塞站点将来自不同接入树的数据包入不同的虚拟输出队列, 再对这些虚拟输出队列采用 DRR 公平调度保证接入树在拥塞链路处公平分享带宽. 该算法采用非线性控制来减少业务量振荡, 提高稳定性. 但由于该算法要求拥塞站点为所有接入树设置输出队列, 同时在它们之间进行公平调度, 大大增加了站点的处理负荷, 这使得该算法在高速 RPR 网中难以实现. 文献[7]中提出了基于反馈拥塞机制的 REDUCE 算法: 上游站点在拥塞发生时不会直接将本地限速速率降为接收到的反馈速率, 拥塞消失时也不会将本地限速速率直接提升至链路速率, 而是根据下游站点的拥塞程度来逐渐地增加或者减少本地限速速率. 由于该算法对本地限速速率的变化幅度缺乏控制, 所以当出现非平衡流时振荡依然严重. 文献[8]为了缩短收敛时间提出了基于拥塞触发机制的公平算法, 当站点检测到拥塞发生就立刻广播拥塞控制消息. 文献[9]在 RPR 草案的基础之上提出了避免 HOL 问题的改进算法. 然而上述这些算法都不能很好地解决高速 RPR 网络中存在的所有问题, 本文将提出一种更加适用的公平算法. 本文的组织结构如下: 在下一部分中对线性逼近最佳公平速率(LAOFR))) Linear Approach Optimal Fair Rate) 的算法作详细的描述和分析. 第三部分是算法的仿真和对仿真结果的讨论. 最后是文章的总结和对今后工作的展望.

2 LAOFR 算法

本文的公平算法遵循 RIAS 公平性原则[10]. RIAS 公平性原则不同于传统的最大最小公平性[11]和按比例公平性[12], 能在基于站点和基于流的两种粒度上保证公平. 由于 RPR 中的公平控制主要指的是站点之间的公平带宽分配问题, 因此本文讨论的是基于站点粒度的公平算法, 本文提出了一种线性逼近最佳速率的公平算法))) LAOFR 算法. 它能保证 RPR 环上的站点公平分享链路上的可用带宽. 同时当环上存在非平衡流时也不会出现流量振荡. RPR 草案中的算法造成流量振荡的原因是当拥塞出现时上游站点直接用接收到的反馈控制速率作为本地限速速率; 而当拥塞消失时上游站点的本地限速速率又逐渐增加, 直至环路上重新出现拥塞. 随着拥塞的不断出现与消失, 上游站点的本地限速速率也随之不断在反馈控制速率和链路速率之间变化, 从而出现剧烈振荡. 而 LAOFR 算法借鉴了文[7]的做法, 并不直接将反馈控制速率作为本地限速速率, 而是控制本地限速速率逐渐逼近一个最佳的公平速率, 此公平速率符合 RIAS 公平性原则. 由于算法最终能够收敛到稳定状态, 因此不会出现流量振荡.

为了较好地描述算法, 我们首先定义两种站点状态: 拥塞状态: 如果拥塞发生在下游站点, 那么上游站点的状态定义为拥塞状态.

非拥塞状态: 如果下游站点都不存在拥塞, 那么上游站点的状态定义为非拥塞状态.

2.1.1 算法的具体描述

step1 算法初始化: 所有站点均设定两个速率阀值%%低速率阀值 LowThreshold 和高速率阀值 HighThreshold. LowThreshold 初始化为 0, HighThreshold 初始化为链路速率. 本地限速速率 allowedRate 的初始值等于 HighThreshold.

step2 当拥塞发生在下游站点时(拥塞状态), 上游站点的 allowedRate 将按照 2.2 中的线性经验公式逐渐降低. 这是一个 allowedRate 向 LowThreshold 逼近的过程, 此过程将持续下去直至下面 a, b 两种情况之一发生:

a. 下游站点拥塞消失(非拥塞状态), 算法转入 step3;

b. 上游站点的 allowedRate 已经非常接近 LowThreshold, 算法转入 step5;

Step3 当上游站点从拥塞状态转换到非拥塞状态时, 首先将 LowThreshold 置为当前的 allowedRate, 由于现在处于非拥塞状态, 说明下游链路上还有可用带宽, 所以上游站点的 allowedRate 又按照 2.2 中的线性经验公式逐渐增加, 这是一个 allowedRate 向 HighThreshold 逼近的过程, 此过程将持续下去直至下游站点重新出现拥塞(拥塞状态), 然后算法进入 step4.

Step4 当上游站点从非拥塞状态转换到拥塞状态时, 首先将 HighThreshold 置为当前的 allowedRate, 此时由于站点重新回到拥塞状态, 因此算法又回到 step2.

Step5 算法已经收敛到稳定状态, 算法结束.

从上面的算法描述中可以看到站点的本地限速速率将始终被控制在低阀值与高阀值之间, 即:

LowThreshold < allowedRate < HighThreshold 算法的伪代码将在表 2 中给出.

2.1.2 算法分析

算法只用接收到的反馈速率作为判断拥塞是否出现的依据, 并不直接将它作为本地的限速速率. 即如果接受的反馈速率小于链路速率, 则判断出现了拥塞, 如果接受的反馈速率等于链路速率, 则判断未出现拥塞. 每一次的拥塞状态向非拥塞状态的转换都会使低阀值提升到当前的本地限速速率, 而每一次的非拥塞状态向拥塞状态的转换都会使高阀值减低到当前的本地限速速率, 这样经过几次有限的状态转移, 高阀值和低阀值会逐渐相互逼近, 最终低阀值, 高阀值, 本地限速速率会收敛到同一个值, 这个值即为站点的公平速率. 本地限速速率的递增和递减过程是由一个线性经验公式来控制的. 表 1 给出了此线性经验公式的表达式.

表 1 线性经验公式

Increase: $X = Xc + (HighThreshold - Xc)/Y$
Decrease: $X = Xc - (Xc - LowThreshold)/Z$

其中 Xc 为本地限速速率, X 为新计算出来的本地限速速率. Y 为递增控制因子, Z 为递减控制因子.

在 2.1 中的算法描述中, 我们把 step2y step3y step4 的三个步骤称为一次迭代过程, 每次迭代都会导致高, 低阀值发生变化. 设第 i 次迭代后得到的高阀值为 H_i, 低阀值为 L_i, 则高阀值与低阀值之间的差值 $v_i = H_i - L_i$. 现在考察第 i+1 次迭代, 设在第 i+1 次迭代中算法由 step2y step3 本地限速速率需要 N 次递减, 每次递减后得到的本地限速速率为 A_K, k = 1, 2, , , N. 由 step3y step4 本地限速速率需要 M 次递增, 每次递增后得到的本地限速速率为 B_K, k = 1, 2, , , M. 第 i+1

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

次迭代后得到的高阀值为 H_{i+1}, 低阀值为 L_{i+1}, 差值 $v_{i+1}=H_{i+1}-L_{i+1}$. 结合线性经验公式我们可以得出以下的递推方程式:

$A_1=H_i$

$A_2=A_1-(A_1-L_i)/Z$

, ,

$A_K=A_{K-1}-(A_{K-1}-L_i)/Z$

, ,

$A_N=A_{N-1}-(A_{N-1}-L_i)/Z$

由上述方程式 $A_2-L_i=A_1-(A_1-L_i)/Z-L_i=(A_1-L_i)(1-1/Z)$, 因为 $A_1=H_i>L_i$, $Z>1$, 所以 $A_2-L_i>0$, $A_2>L_i$. 同理根据 $A_{K-1}>L_i$可推出 $A_K>L_i$, 由 $A_{N-1}>L_i$ 可推出 $A_N>L_i$. 因为低阀值会置为 N 次递减后的本地限速速率, 即 $A_N=L_{i+1}$, 所以 $L_{i+1}=A_N>L_i$.

$B_1=L_i$

$B_2=B_1+(H_i-B_1)/Y$

, ,

$B_K=B_{K-1}+(H_i-B_{K-1})/Y$

, ,

$B_M=B_{M-1}+(H_i-B_{M-1})/Y$

由上述方程式 $H_i-B_2=H_i-B_1-(H_i-B_1)/Y=(H_i-B_1)(1-1/Y)$, 因为 $B_1=L_i<H_i$, $Y>1$, 所以 $H_i-B_2>0$, $B_2<H_i$. 同理根据 $B_{K-1}<H_i$可推出 $B_K<H_i$, 由 $B_{M-1}<H_i$可推出 $B_M<H_i$. 因为高阀值会置为 M 次递增后的本地限速速率, 即 $B_M=H_{i+1}$, 所以 $H_{i+1}=B_M<H_i$. 因为 $L_{i+1}>L_i$, $H_{i+1}<H_i$, 且 $v_{i+1}=H_{i+1}-L_{i+1}$, $v_i=H_i-L_i$. 最终得出结论 $v_{i+1}<v_i$, 即算法每经过一次迭代, 高低阀值之间的差距就会减少, 这样随着高阀值和低阀值的相互逼近, X 最终会收敛到一个稳定的最佳值.

算法达到稳定状态后, 如果环上的流量发生变化(比如出现新的站点竞争带宽或者站点降低自身的发送速率等) , 则站点的高, 低阀值要作相应调整, 然后重新收敛到一个新的稳定状态以适应这种变化. 因此站点在达到稳定状态后还需要监控环上的流量变化, 并根据监控结果调整高阀值或低阀值, 由于高阀值或低阀值发生了变化, 因此原有的稳定状态条件: 高阀值= 低阀值= 本地限速速率将不再成立, LAOFR 算法将在调整后的阀值的基础上重新收敛到一个新的稳定状态. 限于篇幅, 流量监控的算法将在后续文章中给出.

LAOFR 算法与算法[7] 最大的不同在于算法引入了两个阀值来控制本地限速速率变化的幅度, 通过阀值的相互逼近导致变化幅度的迅速减少, 从而最终能够收敛到稳定状态. 而算法[7] 和 RPR 草案[3] 都缺乏对变化幅度的控制, 所以在非平衡流的情况下会永久振荡.

复杂度分析: 算法只用到了两个额外的阀值变量和两个控制因子, 因此空间复杂度为 O(1) ; 算法的时间复杂度为 O(K(M+N)) , 其中 K 为总的迭代次数, M 为每次迭代过程中递增次数的最大值, N 为每次迭代过程中递减次数的最大值. 由于没有额外的存储开销, 因此算法的复杂度优于[4] [5] 中的算法复杂度.

表 2　算法的伪代码

```
Init: HighThreshold= LINK_RATE;
      LowThreshold= 0;
    While(true) {
      if( downstream station exists congestion) {
        if( state= = non_congested) {
          HighThreshold= allowedRate;
          state= congested;
        }
          allowedRate= allowedRate2(allowedRate2lowThreshold)/Z;
          if( allowedRate U lowThreshold)
                exit;
          }
      else{
              if ( state= = congested) {
          state = non_congested;
          lowThreshold = allowedRate;
          }
          allowedRate = allowedRate + (HighThreshold2allowe2
dRate)/Y;
        }
    }
```

3　仿真结果和分析

本节将通过仿真来分析 LAOFR 算法在高速 RPR 网络中的性能. 我们首先比较了在非平衡流环境中 LAOFR 算法与 RPR 草案[3] 的性能. 然后仿真并分析了 LAOFR 算法在 Hot2Receiver[3] 环境中的性能. 表 3 中列出了仿真参数, 采用的仿真软件是 OPNET8. 0, 仿真拓扑如图 2 所示.

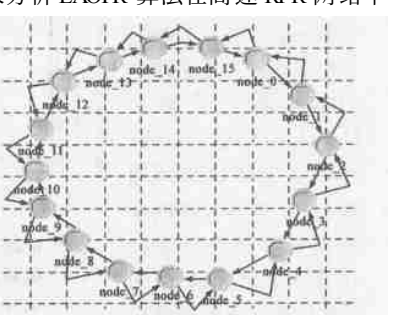

图 2　RPR 仿真拓扑

表 3　仿真参数

链路速率(OC2192)	10Gbps/ 秒
转发队列长度(单对列模式)	256 kBytes
链路传播延时	70 微秒
仿真节点数	16
公平算法计算时间间隔	0. 1 毫秒
公平控制包广播时间间隔	4. 5 微秒
递增控制因子 Y	32
递减控制因子 Z	32

311　非平衡流环境下 LAOFR 与 RPR 草案算法性能比较

仿真场景 1: node_3 以 10Gbps/ 秒的平均速率从外环向 node_1 发送数据(Flow[3, 1]) , node_2 以 1Gbps/ 秒的平均速率

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

同时从外环向 node. 1 发送数据(Flow[2, 1]). 公平控制采用 RPR 草案算法. 图 3 中显示了 Flow[3, 1], Flow[2, 1]的流量变化. 从仿真结果可以看到: 草案算法在非平衡流环境下存在剧烈的振荡.

结果分析: 由于 node. 1 与 node. 2 之间的外环链路(Link[2, 1])负荷超过了链路容量(Flow[3, 1] + Flow[2, 1] = 11Gbps/秒 > 10Gbps/秒), 所以 node. 2 检测到拥塞并向上游站点(node. 3)广播公平控制消息, 控制消息中的反馈速率为 node. 2 的本地发送速率(1Gbps/秒). 按照 RPR 草案中的规定, node. 3 在接收到公平控制消息之后, 本地限速速率和本地发送速率被降为 1Gbps/秒. 此时 Link[2, 1]的链路负载也随之降为 2Gbps/秒(flow[3, 1] + flow[2, 1] = 2Gbps/秒), 从而 node. 2 检测到拥塞消失, 这将导致 node. 3 的本地限速速率提升为链路速率(10Gbps/秒), 结果是 Link[2, 1]又重新过载, node. 2 重新检测到拥塞, node. 3 的本地限速速率又下降为 1Gbps/秒. 如此往复, node. 3 的本地限速速率将始终在 1Gbps/秒与 10Gbps/秒之间变化, 导致 flow[3, 1]的流量变化呈现剧烈振荡.

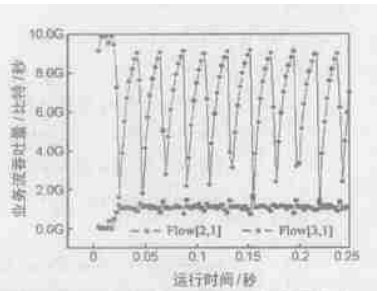

图 3 Flow[3,1]和 flow[2,1]的流量变化(RPR 草案算法 非平衡流环境)

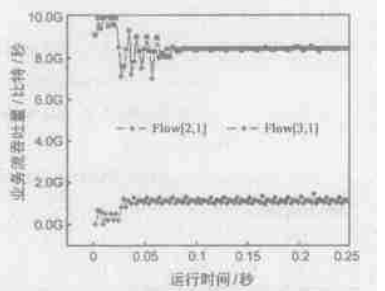

图 4 Flow[3,1]和 flow[2,1]的流量变化

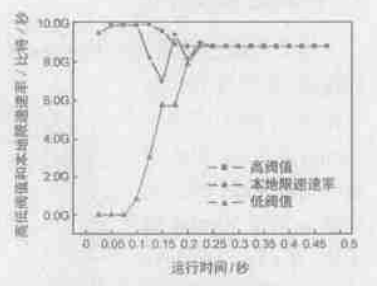

图 5 node_3 的本地限速速率, 高, 低阀值(LAOFR 算法非平衡流环境)

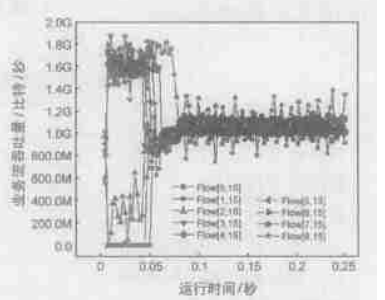

图 6 Flow[0,15]-Flow[8,15]的吞吐量变化(LAOFR 算法, Hot-Receiver 环)

仿真场景 2: 除了公平控制算法采用 LAOFR 算法之外, 其他的仿真条件与仿真场景 1 完全相同. Flow[3, 1]与 Flow[2, 1] 的流量变化显示在图 4 中. node. 3 的本地限速速率, 高低阀值的变化显示在图 5 中. 从仿真结果可以看到: LAOFR 算法成功避免了非平衡流问题.

结果分析: 当 node. 2 检测到拥塞时, 它向上游站点 node. 3 广播反馈控制速率(1Gbps/秒). 根据 LAOFR 算法, node. 3 在接收到此反馈速率后, 它的本地限速速率不会直接置为 1Gbps/秒, 而是从高阀值(10Gbps/秒)逐渐降低直至 node. 2 检测到拥塞消失. 此时 node. 3 的低阀值被置为当前的本地限速速率(≈5.9Gbps/秒), 然后本地限速速率由于拥塞消失而开始逐渐上升直至 node. 2 重新检测到拥塞, 此时 node. 3 的高阀值被置为当前的本地限速速率(≈9.6Gbps/秒), 经过几次往复后, node. 3 的高阀值与低阀值的差值逐渐减小, 本地限速速率的振荡幅度也随之逐渐减小, 最终高, 低阀值, 本地限速速率在很短的时间(≈0.15s)内收敛到一个稳定的值(≈8.8Gbps/秒).

3.2 Hot-Receiver 环境下 LAOFR 算法的性能

9 个站点(node. 0 - node. 8)以 1.5Gbps/秒的平均速率同时从外环向 node. 15 发送数据, 公平控制算法采用 LAOFR 算法. Flow[0, 15] - Flow[8, 15]的流量变化显示在图 6 中.

仿真结果与分析: 由于 node. 0 与 node. 15 之间的外环链路 Link[0, 15]是负载最重的链路, 所以 node. 0 成为 Hot-Receiver, Link[0, 15]成为拥塞链路. LAOFR 算法控制 Flow[0, 15] - Flow[8, 15]在拥塞链路处公平共享带宽. 仿真显示所有流都快速收敛到同一个稳定值(收敛时间≈0.07s). 这个值(≈1.1Gbps)是公平的, 它相当于链路速率除以流经拥塞链路 Link[0, 15]的流数目(10Gbps/9≈1.1Gbps/秒).

3.3 结论

从仿真结果可以看出 LAOFR 算法是一个适用于高速 RPR 网络环境下的性能良好的公平控制算法. 它能有效地控制本地限速速率的变化幅度. 在各种环境下(非平衡流或 Hot-Receiver), 算法都能达到快速的收敛时间和很小的振荡幅度. 最终的收敛结果也满足 RIAS 公平性原则的要求.

4 结束语

本文提出了一种基于高速弹性分组环的公平控制算法%线性逼近最佳公平速率 LAOFR 算法, 该算法既能消除 RPR 草案无法解决的非平衡流问题, 同时也能很好地实现站点之间公平分享带宽. 仿真结果证明算法具有快速收敛时间, 良好的性能和低复杂度. 在今后的工作中, 我们希望对该算法在多拥塞情况下的性能展开研究.

参考文献:

[1] F E Ross. Overview of FDDI: The Fiber Distributed Data Interface[J]. IEEE Journal on Selected Areas in Communications, 1989, 7(7): 1043 - 1051.

[2] IEEE Standard 802.5- 1989, IEEE standard for token ring[S].

[3] IEEE Draft P802.17/D2.1, IEEE draft for Resilient Packet Ring[S].

[4] Kong Hongwei, Ge Ning, Ruan Fang, Feng Chongxi. A congestion-point oriented congestion control algorithm for resilient packet ring[J]. Chinese Journal of Electronics, 2003, 12(1): 1- 5.

[5] Peng Yue, Zengji Liu, Jing Liu. High performance fair bandwidth allocation algorithm for Resilient Packet Ring[A]. IEEE AINA 03[C].

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

Xi. an, China: 2003. 415.

[6] A K Parekh, R G Gallager. A Generalized Processor Sharing approach to flow control in integrated services networks: The single node case [J]. IEEE/ACM Transactions on Networking, 1993, 2(2): 137- 150.

[7] Stein Gjessing, Fredrik Davik. Performance evaluation of back2pressure fairness in RPR [DB/OL]. http://www.simula.no/photo/paper.pdf, 2002.

[8] Stein Gjessing, Arne Maus. A fairness algorithm for high2speed net2works based on a Resilient packet Ring architecture[A]. IEEE Inter2national Conference on Systems, Man and Cybernetics[C]. Hammamet Tunesia, 2002. 279- 284.

[9] S Gjessing, B F Davik. Avoiding Head of Line blocking using an en2hanced fairness algorithm in a Resilient Packet Ring [A]. International Conference on Telecommunication(ICT)[C]. Beijing, China: 2002. 21 - 26.

[10] Edward W Knightly. RIAS fairness reference model[DB/OL]. http://www. ece. rice. edu/networks/RIAS/, 1998.

[11] Bo. Radunovi, Jean2Yves Le Boudec. A unified packet work for Max2Min and Min2Max fairness with applications[R]. Lausanne, Switzer2land: Technical report IC2200248, 2002.

[12] F Kelly. Charging and rate control for elastic traffic[J]. European Transactions on Telecommunication, 1997, 8: 33- 37.

作者简介:

柳立峰　男, 1974 年 4 月出生于河南省平顶山, 现为北京邮电大学网络与交换国家重点实验室在读博士生, 研究方向: 宽带通信网络, 移动计算技术等. E2mail: llfxx@263. net.

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net