

文章编号: 1001-0920(2005)04-0478-03

## 主动队列管理中 RQC 控制器的设计

蔡小玲<sup>1</sup>, 汪小帆<sup>2</sup>, 王执铨<sup>1</sup>

(1. 南京理工大学 自动化系, 江苏 南京 210094; 2 上海交通大学 自动化系, 上海 200030)

**摘要:** 基于网络中输入输出速率和队列长度均可帮助决定更精确丢弃概率的思想, 提出了根据输入输出速率和队列长度决定包的丢弃标注概率的 AQM 算法, 即 RQC 算法. 通过仿真将该算法与 RED 和 PI 算法进行比较, 说明了 RQC 控制算法的优点.

**关键词:** RQC; 主动队列管理; RED; 队列长度; 输入输出速率

**中图分类号:** TP391

**文献标识码:** A

### RQC controller for active queue management

CAI Xiao-ling<sup>1</sup>, WANG Xiao-fan<sup>2</sup>, WANG Zhi-quan<sup>1</sup>

(1. Nanjing University of Science and Technology, Department of Automation, 210094 Nanjing, China;

2 Department of Automation, Shanghai University of Jiaotong, Shanghai 200030, China Correspondent: CAI Xiao-ling, E-mail: cxl-hello@163.com)

**Abstract:** Based on the idea that both input/output rate and queue length can help the determination of the accurate drop probability in the network, an algorithm called rate/queue controller (RQC) is presented. The method computes drop probability based on input/output rate and queue length. This algorithm is compared with random early detection (RED). The simulation results show some features of RQC algorithm.

**Key words:** RQC; AQM; RED; queue length; input/output rate

## 1 引言

Internet 路由器中的主动队列管理机制 (AQM) 可通过提前检测拥塞来提高网络的吞吐量, 减少响应时间, 并提高数据流之间的公平性. 网络拥塞控制是近年来的研究热点之一<sup>[1]</sup>, RED<sup>[2]</sup> 是一种典型的 AQM 算法, 其主要目的是在缓存溢出之前将包丢弃, 提前通知源端拥塞即将出现, 避免 TCP 的同步和数据流中的不平衡, 对不支持相应机制的源端采取一定的措施.

自 RED 出现后, 相继又出现了 RED 的变化版本, 包括: FRED<sup>[3]</sup> (Flow RED), SRED<sup>[4]</sup> (Stabilized RED) 和 BLUE<sup>[5]</sup>. 尽管 RED 及其变化版本通过简单的队尾丢弃提高了网络的性能, 但仍存在两方面缺点: 一是参数难以正确设置; 二是它对实时的队列

到达和离开的速率不敏感, 而是依赖于长期的平均队列长度. 利用 REM<sup>[6]</sup> (Random exponential marking), AVQ<sup>[7]</sup> (Adaptive virtual queue) 和 PI<sup>[8]</sup> (Proportional Integral) 算法设计的控制器, 虽然解决了系统对输入/输出速率不敏感问题, 但它们并未考虑队列长度对网络性能的影响. 还有一些算法实现起来较为复杂.

本文针对上述问题, 提出一种根据队列长度和输入/输出速率计算丢弃概率的主动队列算法 RQC, 并将 RQC 与主动队列算法 RED 及 PI 算法在响应时间、队列长度和通信量方面进行比较. 仿真结果显示了本文算法的特性和优点.

## 2 基于队列和速率的 AQM

### 2.1 基于队列的 AQM

收稿日期: 2004-06-21; 修回日期: 2004-08-05

基金项目: 教育部博士点基金项目 (20020288025); 江苏省自然科学基金项目 (BK2004132).

作者简介: 蔡小玲 (1977—), 女, 江苏常州人, 博士生, 从事控制理论、控制工程的研究; 王执铨 (1939—), 男, 湖北武汉人, 教授, 博士生导师, 从事混沌控制、动态大系统的研究.

RED, FRED 和 SRED 均属于基于队列的 AQM 类 RED 利用长时间的平均队列长度吸收暂时的突发数据流量, 在 RED 中队列长度是唯一的拥塞指示器 RED 假设队列长度间接反映网络负载和队列中包离去的快慢情况, 因此它不会对网络变化立刻作出相应的反应 网络建立的初期将队列长度调整到稳定队列层次的时间也较长, 而且它的参数较多, 难以设置到合适的数值 若参数设置不恰当, 则性能表现不及最简单的 DT (丢尾), 所以目前 RED 仍不能在网络中实际运行

## 2.2 基于速率的 AQM

将速率信息用于 REM, AVQ 和 PI<sup>[8]</sup> 控制器等, 可提高响应速度, 但必须小心利用以避免网络系统不稳定 采用速率信息的优点参见文献[8, 10] 这些机制中利用不同的方法检测速率信息, 如队列长度变化率, 或输入速率与输出速率之差 本文采用一个简单的速率指示器作为负载因子, 定义为周期测量的包到达速率(输入)和服务速率(输出)的比值

$$\text{Rate Factor} = \frac{\text{Input Rate}}{\text{Output Rate}} \quad (1)$$

其中: Input Rate 为输入速率, 即链路中包进入队列的速度; Output Rate 为输出链路容量

如果负载因子的值较大, 说明有拥塞现象存在; 值较小, 则说明链路的利用率不够 上述两种情况都是不希望发生的, 路由器应将负载因子维持在小于 1 (接近 1) 的稳定状态

采用负载因子后, 队列长度的信息仍很重要, 因为队列长度是更稳定(比较慢的)的拥塞指示器, 而且在计算丢弃概率时, 队列长度信息可能是有用的, 它允许队列中包的离开存在滞后 这保证了小的队列延迟, 使得剩下的缓存可吸收包的突然流量

## 3 速率/队列控制(RQC)

本文提出的主动队列控制算法 RQC 不但保持了 RED 的优点, 即惩罚不合规的数据流, 避免全局同步化, 而且具有更好的负载(输入/输出速率)响应, 并提供了简单直接的参数设置

RQC 的目的是: 1) 将队列长度维持在期望的队列水平上; 2) 将队列中包的到达速率维持在队列服务速率附近(小于等于). 在到达速率、网络带宽、队列长度和期望队列长度可知的情况下, 设计 RQC 控制器

### 3.1 测量负载因子和队列长度

队列中每进入 1 个包, RED 就利用指数加权平均方法(EWMA)计算 1 次平均队列长度 计算平均队列长度的目的是为了减少过多偏向于流量多的数据流 RQC 每过 1 个采样时间  $\tau$  便重新计算 1 次负载

因子, 而不是每到 1 个包计算 1 次, 但其队列长度仍根据包进入队列的情况随时更新 在该实验中, 负载因子取平均输入/输出速率, 这样可避免对时间间隔  $\tau$  的敏感性, 它强调当前采样值以提高算法对当前通信负载的响应速度 队列长度取为实时的队列长度

### 3.2 丢失概率的计算

RQC 分别计算负载因子和队列长度与其期望值之差 如果差值均为 0, 则意味着该拥塞指示器达到了目标值 两个差值的各自指数值相加后, 再加上上一个丢弃概率的值为包的丢弃概率 采用加法的原因是可将这两个差值分开计算, 这样更容易增加其中一个差值的比重 RQC 计算丢弃概率如下:

$$P_{\text{drop}} = (1/4) \min(1, (R_{\text{avg}} - R_{\text{ref}})^a) + (3/4) \min(1, (Q_{\text{avg}} - Q_{\text{ref}})^b) + P_{\text{old}} \quad (2)$$

达到理想稳定状态时, 负载因子和平均队列长度都达到了期望值, 当前的丢弃概率等于上一次的丢弃概率 式(2)中: 1/4 和 3/4 为乘数因子, 它们的改变取决于需要增加哪个差值的比重; 取参数  $R_{\text{ref}} = 0.95$ ;  $Q_{\text{ref}}$  可根据仿真结果选择一个比较合适的值;  $R_{\text{avg}}$  和  $Q_{\text{avg}}$  分别是输入速率、输出速率及平均队列长度;  $P_{\text{old}}$  为上一个采样时间的丢弃率

## 4 性能评估

采用 RQC 算法在仿真器 ns 上进行仿真<sup>[9]</sup> 将 RQC, PED 和 PI 算法同时运行在主链路的路由器上, 并比较 3 种算法的响应时间、队列长度和通信量

RED 算法的参数设置为:  $m_{\text{in\_th}} = 150$ ,  $m_{\text{ax\_th}} = 700$ ,  $\omega_q = 0.002$ ,  $m_{\text{ax}_p} = 0.1$ ; RQC 算法的参数设置为:  $Q_{\text{ref}} = 300$ ,  $R_{\text{ref}} = 0.95$ ,  $\omega_q = 0.002$ ,  $a = 6$ ,  $b = 5$ ,  $\tau = 0.002$  s; PI 算法的参数设置为:  $Q_{\text{ref}} = 300$ , 其他参数取默认值

### 4.1 仿真模型

仿真拓扑图如图 1 所示。

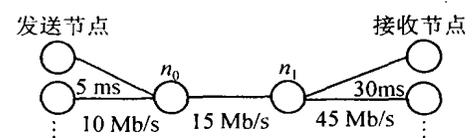


图 1 仿真拓扑图

网络仿真的设置为: 数据源为 360 个 FTP; 发送节点与  $n_0$ ,  $n_0$  与  $n_1$  之间的带宽分别为 10, 15 Mb/s, 延迟均为 5 ms;  $n_1$  与接收节点之间的带宽为 45 Mb/s, 延迟为 30 ms; 平均队列长度设为 500 字节; 缓存大小为 800 个包; 平均拥塞窗口大小设为 15 个包 仿真共运行 60 s

## 4.2 仿真结果

RED, RQC 和 PI 的仿真队列长度分别如图 2~图 4 所示。表 1 为 3 个算法的响应时间和通信量的数据比较。从图 2~图 4 可以看出, 采用 RED 算法时, 队列达到稳定队列长度所需的时间大约是 15 s, PI 算法需要约 23 s, 而新算法只需约 4 s 的时间。并且在运行过程中 RED 队列长度的变化比较剧烈, 因为它根据丢失概率的不同在不断调整其长度; PI 的队列长度变化更大, 因为它仅根据实时队列长度计

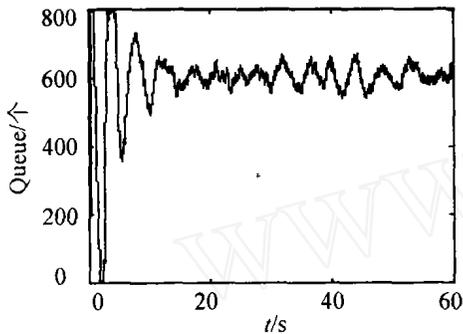


图 2 RED 的仿真队列

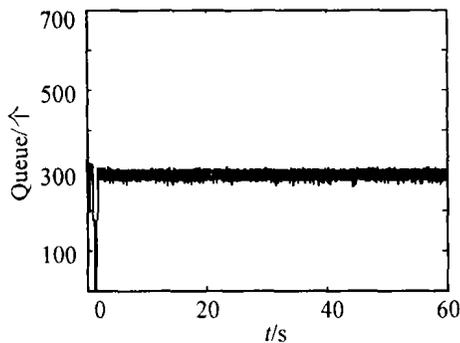


图 3 RQC 的仿真队列

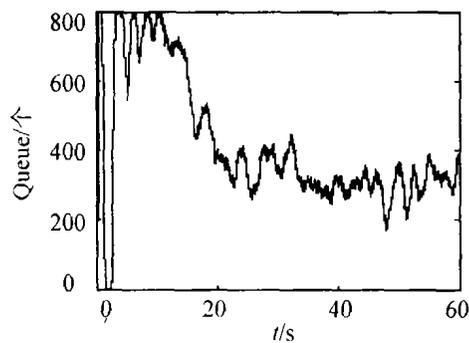


图 4 PI 的仿真队列

表 1 RED, PI 和 RQC 的仿真数据比较

|     | 响应时间 /ms | 通信量 /(bit/s) |
|-----|----------|--------------|
| RED | 323.107  | 1.838.832    |
| RQC | 232.686  | 1.856.589    |
| PI  | 282.835  | 1.841.991    |

算包的丢弃率; 而 RQC 的队列长度的变化非常小, 稳定地运行在队列长度附近。

从表 1 也可看出, 无论是平均响应时间还是通信量, RQC 的性能都优于 RED 和 PI。

## 5 结论

通过仿真将本文的 RQC 与 RED 及 PI 算法进行比较。基于队列的 RED 算法, 尤其是 FRED, 性能较好, 但难以构造满意的参数; 基于速率的 PI 算法减少了队列延迟, 但不使用 ECN<sup>[9]</sup>时, 通信量较低; 而结合了负载和队列长度来计算丢弃概率的 RQC 算法, 降低了响应时间, 提高了通信量, 而且使用者可随意选择增加队列或速率在计算中的比重。

该算法需要进一步解决的问题是研究 RQC 采用 ECN 后能否产生不同的性能, 及其在不同网络环境中是否有不同的性能表现。

## 参考文献 (References)

- [1] 章森, 吴建平, 林闯. 互联网端到端拥塞控制研究综述 [J]. 软件学报, 2002, 13(3): 1-10  
(Zhang M, Wu J P, Lin C. Survey on internet end-to-end congestion control [J]. *J of Software*, 2002, 13(3): 1-10)
- [2] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance [J]. *IEEE/ACM Trans on Networking*, 1993, 1(4): 397-413
- [3] Lin D, Morris R. Dynamics and random early detection [A]. *Proc of the ACM Sigcomm [C]*. Cannes, 1997: 127-136
- [4] Ott T J, Lakshman T V, Wong L H. SRED: Stabilized RED [A]. *Proc of the IEEE Infocam [C]*. New York, 1999: 1346-1355
- [5] Feng W, Kandlur D, Saha D, et al. BLUE: A new class of active queue management algorithms [R]. Michigan: University of Michigan, 1999.
- [6] Lapsley D E, Low S H. Random early marking for internet congestion control [A]. *Proc of the IEEE Globecom [C]*. Rio de Janeiro, 1999: 1747-1752
- [7] Kunniyur S, Srikant R. End-to-end congestion control: Utility functions, random losses and ECN marks [A]. *Proc of the IEEE Infocam [C]*. Tel Aviv, 2000: 1323-1332
- [8] Holot C V, Misra V, Towsley D, et al. On designing improved controllers for AQM routers supporting TCP flows [A]. *Proc of the IEEE Infocam [C]*. Aaska, 2001: 1726-1734
- [9] Ramakrishnan K, Floyd S. A proposal to add explicit congestion notification (ECN) to IP [J]. *Request for Comments 2481*, 1999.
- [10] Holot C V, Misra V, Towsley D, et al. A control theoretic analysis of RED [A]. *Proc of the IEEE Infocam [C]*. Aaska, 2001: 1510-1519.