

基于智能预测控制的网络拥塞主动 队列管理算法研究

陈增强,刘忠信,袁著祉,王繁珍

(南开大学 自动化系,天津 300071)

摘要:路由队列管理是保证网络性能、避免网络 congestion 的重要手段,目前采用的主要队列管理方法为被动式队列管理,同时主动式队列管理已经成为近来的主要研究热点.随机早检测 (RED) 作为最早提出的主动队列管理方法,更获得了普遍的关注.使用严格的数学模型来描述由端系统和网关组成的系统,并进行队列管理性能分析.提出一种采用快速广义预测控制的 RED 控制器 (FGPC-RED 控制器),进行网络拥塞控制的研究.介绍了系统的结构及系统的辨识,并通过仿真证明了 FGPC 算法在路由队列管理中应用的可行性,可以有效控制队列长度,避免路由 congestion 及减小往返延迟.

关键词:智能预测控制;网络拥塞控制;主动队列管理;网络模拟器;RED 算法

中图分类号: TP273.2 **文献标识码:** A **文章编号:** 1673-4785 (2008) 04-0313-08

An active queue management algorithm for network congestion based on intelligent prediction control

CHEN Zeng-qiang, LIU Zhong-xin, YUAN Zhu-zhi, WANG Fan-zhen

(Department of Automation, Nankai University, Tianjin 300071, China)

Abstract: Queue management is an important means to improve the performance of a network and avoid network congestion. At present, passive queue management dominates the queue management field. Recently active queue management (AQM) has been getting more and more attention, along with the random early detection (RED) algorithm, the first proposed AQM algorithm. In order to analyze a system made up of clients and a gateway, a strict mathematical model was developed to describe the system. A kind of RED controller with fast generalized prediction control was included to analyze congestion control. The system's architecture and identification are discussed in detail. A series of simulations were made, showing that fuzzy generalized predictive control (FGPC) is feasible in queue management and effective in controlling queue length, avoiding network congestion, and reducing round trip delay.

Keywords: intelligent predictive control; network congestion control; active queue management; network simulator; RED algorithm

计算机网络在过去的十几年中经历了爆炸式的增长,特别是目前迅速发展的多媒体通信业务,进一步推动 Internet 规模的扩大和业务的增长.但随着

计算机网络规模和负荷的增加,拥塞问题也越来越严重.同时随着科技的发展,网络设备的处理速度不断加快,网络带宽持续增长,但硬件的发展往往赶不上实际需求的增长,而且局部网络资源不足,也会导致网络拥塞.拥塞导致的直接结果就是分组丢失率提高,端到端时延加大,甚至有可能使整个系统发

收稿日期: 2007-10-08

基金项目: 国家自然科学基金资助项目 (60774088); 教育部科学技术研究重点项目 (107024).

通信作者: 陈增强. E-mail: chenqz@nankai.edu.cn

生崩溃^[1].当网络处于拥塞崩溃状态时,微小的负载增量都将使网络的有效吞吐量急剧下降.当负载减小时,拥塞可以自动恢复.

单纯地增加网络资源之所以不能解决拥塞问题,是因为拥塞本身是一个动态问题,它不可能只靠静态的方案来解决,而是需要协议在网络出现拥塞时保护网络的正常运行.目前对互联网进行的拥塞控制主要是依靠在源端执行的基于窗口的 TCP 拥塞控制^[2]机制. TCP 拥塞控制是一种端到端的控制机制,它是通过改变一些参数实现的. TCP 基于窗口的端到端的拥塞控制对于 Internet 的鲁棒性起到了关键作用.然而,随着 Internet 本身的迅速发展,其网络规模越来越庞大,结构日趋复杂.仅仅依靠端到端的拥塞控制是不够的,网络也需要参与资源的控制工作.路由器直接掌握着互联网上各种网络传输信息,能够有效地监控队列的长度,从而尽早检测到早期的拥塞 (incipient congestion).另外,路由器能够全面地审视各个流对拥塞造成的影响,从而决定将拥塞信息通知哪个源端,使其降低数据发送速度.因此采用基于路由器的拥塞控制和 TCP 端到端拥塞控制相结合的方法,是解决目前互联网拥塞控制问题的一个主要途径^[3].

基于路由器的拥塞控制主要是通过对其队列进行管理来实现的,目前的队列管理主要还是被动式队列管理 (passive queue management, PQM).本质上 PQM 并没有参与到网络拥塞管理中去,只是在队列溢出的情况下被动地丢包.互联网工程工作小组 (internet engineering task force, IETF) 推荐在路由器上使用主动式队列管理 (active queue management, AQM)^[4]机制来控制什么时候丢多少包,从而有效地管理队长度,以支持端到端的拥塞控制.

从控制的观点看,可以将端系统和网关组成的系统看成是一个反馈控制系统.考虑到其复杂性及时延性,为实现对系统的高精度、高可靠以及高鲁棒性的控制效果,需使用具有鲁棒性的预测控制器. Clark 等人提出的广义预测控制 (generalized predictive control, GPC) 是一种基于模型参数的自适应预测控制算法,其

在延迟时间与模型时间不一致时仍具有良好的鲁棒稳定性.另外,这种控制算法基于传统的参数模型,其模型参数少,并采用多步预测、滚动优化和反馈校正等策略,控制效果好^[5].基于以上的因素,文献 [6] 提出了一种快速广义预测控制算法 (fast generalized predictive control, FGPC),与传统 GPC 相比,具有更快的运算速度,满足控制实时性的要求.本文基于文献 [6] 的 FGPC 算法提出了一种网络拥塞控制主动队列管理算法 (FGPC-RED 算法). FGPC-RED 控制方法可有效地减少拥塞现象的发生,降低往返延迟,改善网络的性能.

1 RED 算法

在当前的互联网上,丢包是对端节点进行拥塞通知的主要机制,解决路由器“满队列”问题的方法便是在队列满之前丢包,这样端节点便能在队列溢出前对拥塞做出反应.这种方法便称为“主动式队列管理”.

RED^[7]算法是最常用的一种 AQM 技术,其基本思想是通过监控路由器输出端口队列的平均长度来探测拥塞.一旦发现拥塞逼近,就随机地选择源端来通知拥塞,使它们在队列溢出导致丢包之前降低发送数据速率,从而缓解网络拥塞.这样对面向连接的 TCP 数据流来说,RED 就有可能避免丢弃属于同一连接的连续数据包,从而提高连接的吞吐量.图 1 显示了平均队长与丢包率的关系.

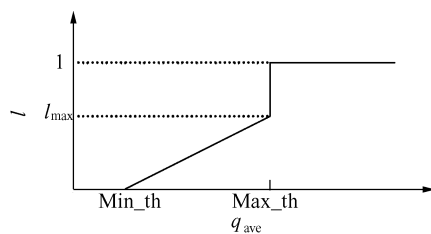


图 1 平均队长与丢包率的关系

Fig 1 The relation between average queue length and packet loss ratio

RED 有 2 个阈值: q_{min} 和 q_{max} . 当一个数据包到达路由器时, RED 将当前 q_{ave} 和这 2 个阈值按以下原则比较: 如果 $q_{ave} < q_{min}$ 时, 将此包排队; 如果 $q_{min} < q_{ave} < q_{max}$, 计算丢弃概率 P , 并以 P 丢弃此包; 如果 $q_{max} < q_{ave}$, 则丢弃此包.

RED的目的是控制拥塞结点的队列长度在最大和最小之间,保持对资源的高利用率. RED 的有效性经过了一些实践的验证,但依旧存在一些缺陷.

1) RED 算法的性能敏感于设计参数和网络状况,在特定的网络负载状况下依然会导致多个 TCP 的同步,造成队列震荡,吞吐量降低和时延抖动加剧.

2) RED 算法的公平性和稳定性也存在问题. 在网络严重拥塞时, RED 往往失去对队列的控制能力,带来的问题甚至比尾部丢弃严重.

3) 自 RED 被首次提出来之后,它的参数配置就是一个没有彻底解决的问题.

正是由于 RED 存在着上面所述诸多问题,导致了其他几种 AQM 技术的产生. 这些 AQM 技术有些是在 RED 基础上的扩充,弥补了 RED 的一些缺点,比如 FRED、ARED 等;另外一些 AQM 算法则是利用链路状态、队列占用状况等信息来管理队列,比如 BLUE、SRED 等.

2 快速广义预测控制 (FGPC) 算法

基于文献 [6] 中的快速广义预测控制方法,对端系统和网关组成的系统采用下面具有随机扰动的离散差分方程来描述:

$$A(z^{-1})y(k) = B(z^{-1})u(k-1) + (z^{-1})/\epsilon. \tag{1}$$

式中: $A(z^{-1}) = 1 + a_1 z^{-1} + \dots + a_{N_a} z^{-N_a}$; $B(z^{-1}) = 1 + b_1 z^{-1} + \dots + b_{N_b} z^{-N_b}$, $\epsilon = 1 - z^{-1}$ 为差分算子, z^{-1} 是后向移位算子, $u(k-1)$ 是系统输入信号, $y(k)$ 是系统输出信号, (k) 是噪声信号.

将式 (1) 两边乘 ϵ 并整理,推导出基于式 (2) 综合加权的目标函数的 GPC 算法:

$$J = \sum_{k=1}^N [p(y_k - w_k)]^2 + \sum_{k=0}^{M-1} (u_k)^2. \tag{2}$$

式中: M 为控制步长, w_k 为 $t+k$ 时刻的期望输出,输出加权 p 和控制加权 λ 分别是 z^{-1} 的多项式.

为避免求解 Diophantine 方程, FGPC 利用反向递推对 GPC 进行变换,算法复杂度仅为 $O(N \cdot n)$, N 与 n 为求解矩阵的维数. 与传统 GPC 相比,具有

更快的运算速度. 正适合了网络流量变化的特点.

算法实现如下:

1) 初始化: 令 $P_N = p$, $R_N = p$, $K_N = 0$, $L_N = 0$, $S_N = 0$, $Q_N = \dots$.

2) 取 $k = N, (N-1), \dots, 1$, 计算:

$$(\bar{P}_k, \bar{K}_k) = (P_k, \hat{A}; K_k, B).$$

即
$$\begin{bmatrix} q^{-1} \bar{P}_k \\ \bar{K}_k \end{bmatrix} = \begin{bmatrix} P_k - \hat{A} \\ K_k + B \end{bmatrix}, \text{ 从而 } \begin{bmatrix} \bar{P}_k \\ \bar{K}_k \end{bmatrix} =$$

$$\begin{bmatrix} q(P_k - \hat{A}) \\ K_k + B \end{bmatrix}, \quad = P_k(0)/\hat{A}(0), (K_{k-1}, Q_{k-1}; P_{k-1},$$

$$L_{k-1}; R_{k-1}, S_{k-1}) = (\bar{K}_k, Q_k; \bar{P}_k, L_k; R_k, S_k),$$

即
$$\begin{bmatrix} q^{-1} K_{k-1} & P_{k-1} & R_{k-1} \\ Q_{k-1} & L_{k-1} & S_{k-1} \end{bmatrix} =$$

$$\begin{bmatrix} \bar{K}_k & \bar{P}_k & R_k \\ Q_k & L_k & S_k \end{bmatrix}.$$

从而
$$\begin{bmatrix} K_{k-1} & P_{k-1} & R_{k-1} \\ Q_{k-1} & L_{k-1} & S_{k-1} \end{bmatrix} =$$

$$\begin{bmatrix} q(\bar{K}_k - Q_k) & \bar{P}_k - L_k & R_k - S_k \\ \bar{K}_k + Q_k & \bar{P}_k + L_k & R_k + S_k \end{bmatrix}.$$

式中: $Q_k = Q_k(0)/\sqrt{\bar{K}_k(0) + Q_k^2(0)}$, $\bar{K}_k = \bar{K}_k(0)/\sqrt{\bar{K}_k(0) + Q_k^2(0)}$.

3) 计算最优控制增量 $u_0 = (S_0 w_r - L_0 y_0)/Q_0$.

4) 计算最优控制量 $u_0 = u_{-1} + u_0$ 转 2).

3 FGPC-RED 控制器的结构

3.1 FGPC-RED 控制器的结构

该控制系统的目标为稳定并控制路由器缓冲区队列长度的大小在一个合适的目标值,来提高链路的利用率及减小队列延迟. 由于 FGPC 控制算法在各个方面的广泛成功应用^[8-10],因此在此采用 FGPC 作为主要的手段来对系统进行控制. 系统控制框图如图 2 所示.

在该系统中, Q_r 为期望的队列长度,是系统的输入设定值, q 为采样时刻的队列长度, q_{ave} 为通过滤波之后得到的平均队列长度, p 为预测所得的丢包率,是被控对象的输出, $w = (z^{-1})/\epsilon$ 为输入系统

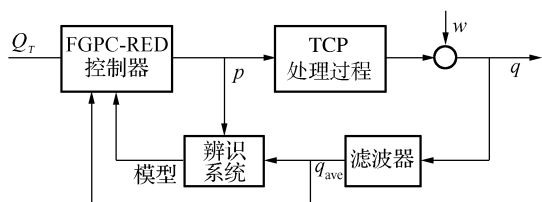


图 2 FGPC-RED 控制系统

Fig 2 FGPC-RED control system

的噪声信号。

将运行时间分成许多个大小固定时间段,即采样间隔为 (t) 。时间标志为 n ,第 n 个时间段用 $T(n)$ 来表示。 $q(n)$ 为在第 n 个时间段的真实队列长度。系统的目标就是找到一个恰当的丢包率使系统的队列长度保持在目标队长,而不受网络的流量及扰动如 UDP 流的影响。

从图 2 可以看出,该系统主要包括 3 个部分:式 (3) 所给的低通滤波器;辨识模块,在该部分采用递推最小二乘法^[11]进行系统辨识;FGPC-RED 控制器,根据 Q_T 对 p 进行预测。

由于网络流量的突变及其他扰动如 UDP 流^[12]的存在,实际的队列长度可能会产生强烈的波动,因此低通滤波器被采用。经过滤波的队列长度为

$$q_{ave}(n) = (1 - \alpha) q_{ave}(n-1) + \alpha q(n). \quad (3)$$

式中: α 为滤波因子,在表现上看来,是平均队列的权。因此,该控制器的主要目标为调整 p ,来控制队列长度到给定的值。

在运行时,应对系统模型进行实时的辨识,此处采用了递推最小二乘法来对系统进行辨识。 \hat{P}_N 为 $N+K$ 次采样数据得到的最小二乘估计量,第 $n+N+1$ 次采样后,得到的估计量为 \hat{P}_{N+1} :

$$\hat{P}_{N+1} = \hat{P}_N + \frac{P_{N+1}}{T_{N+1} P_{N+1} + w_{N+1}^{-1}} \cdot [y(n+N+1) - T_{N+1} \hat{P}_N], \quad (4)$$

$$P_{N+1} = P_N - \frac{P_N T_{N+1} P_N}{T_{N+1} P_N + 1}. \quad (5)$$

各符号和含义请参考文献 [11]。

在 $t(n)$ 时刻,辨识过程为

1) 如果当前辨识的步数 n 小于系统初始设置

的辨识步数 counts

2) 随机产生一个丢包率 p , 为辨识需要, p 在 0.03 ~ 0.06 之间取值;

3) 将 p 赋值给 RED 系统的丢包率;

4) 调用最小二乘法的辨识程序对整个系统进行辨识;

5) 辨识步数 n 加 1。

3.2 FGPC-RED 系统的运算流程及参数配置

FGPC-RED 控制系统在时刻 n 的运行流程为

1) 对当前的队长进行采样,即 $q(n)$;

2) 用式 (3) 计算平均队列长度 $q_{ave}(n)$;

3) 采用递推最小二乘法的式 (4) 与 (5) 对系统的模型进行辨识,并将辨识所得的系统传递给 FGPC-RED 控制器;

4) 通过 FGPC 算法对丢包率 $p(n)$ 进行预测;

5) 在时刻 n 使用 $p(n)$ 作为系统的丢包率;

6) 存储 $q_{ave}(n)$ 及 $p(n)$, 这些将在 $n+1$ 时刻计算时被使用。

在该部分可以看出,FGPC-RED 控制器主要有以下参数:

采样间隔 (t) : 该参数决定了 p 的更新时间间隔。 (t) 减小时,计算量将会增加。另一方面,较小的 (t) 将提高系统的性能。因此,应该合理选取采样间隔,在仿真中 (t) 取为 0.1 s。

目标队列长度 Q_T : 该参数直接决定了系统稳定状态时的队列长度,队列长度将影响使用率及队列延迟。较高的 Q_T 将提高系统的利用率,但会增大队列延迟。该参数应该根据路由器的性能来确定。

滤波因子 α : 该参数影响滤波器的反应速度。较大时,实时的队列长度的变化将较快地影响平均队列长度。常采用的 α 为 0.002。

4 算法仿真与性能评估

对仿真工作的评价,一个重要的标准就是仿真工具的准确性^[12]。在综合比较了多种通信网仿真工具以后,选定了著名的仿真软件 NS 作为仿真平台。NS 为美国国防部远景规划署 (Defense Advanced Re-

search Projects Agency, DARPA) 资助,由 Berkelay 国家实验室开发的网络模拟器 (network simulator).

FGPC-RED 控制器的主要目的是将队列长度控制在一个目标值 Q_T , 并使之具有较快的收敛速度,克服 RED 存在的对参数和网络状况过于敏感,从而造成队列震荡,吞吐量降低和时延抖动加剧现象.

采用如图 3 所示的哑铃状 (dumbbell) 网络拓扑结构,包含 2 个路由器以及多个源节点和目的节点.

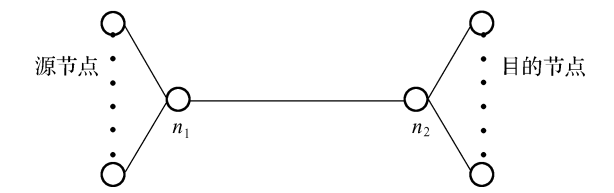


图 3 仿真网络拓扑图

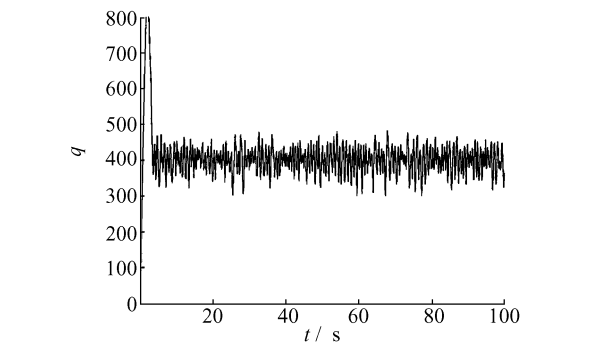
Fig 3 The simulation network topology

如图 3 所示:节点 n_1 与 n_2 为路由器, n_1 与 n_2 间链路的带宽 (capacity) 为 45 Mbps, 传输延迟为 5 ms, 队列 (缓存管理模块的) 类型为 FGPC-RED. 缓冲区的大小为 1 000 个包, 采样间隔 (t) 为 100 ms, 目标队列长度 Q_T 为 400, 滤波因子 为 0.002

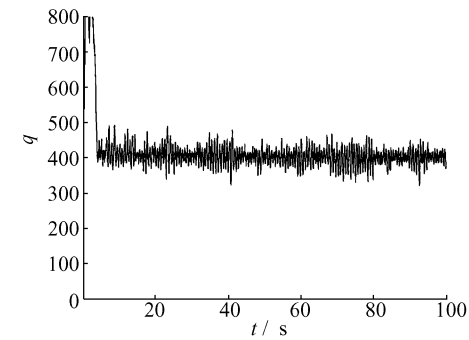
4.1 基于 TCP 连接的仿真

在该部分主要测试是在只有 TCP 连接时, FGPC-RED 控制器是否能达到预期的设计目标.

图 4 显示了 FGPC-RED 控制器在 100 ~ 400 个连接时候的实时队列长度. 可以看到 FGPC-RED 控制器可以比较快地将对列长度控制到并且稳定在 Q_T . 同时可以看出, 400 个连接时平均队长的波动要小于 100 个连接的平均队长的波动.



(a) 100 个 TCP 连接时的情形

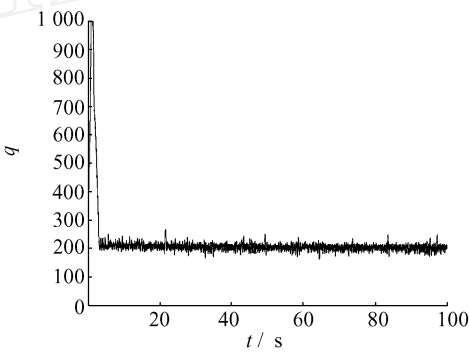


(b) 400 个 TCP 连接时的情形

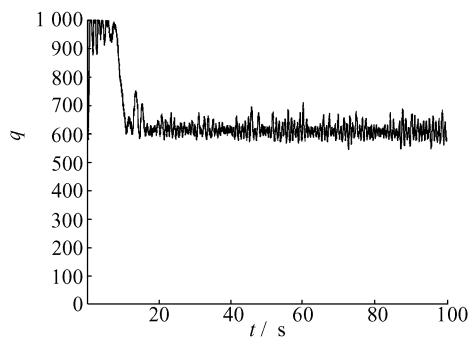
图 4 Q_T 为 400 时, 100 和 400 个 TCP 连接时候的队列长度

Fig 4 The queue length when Q_T is 400 and TCP link is 100 and 400

为了更充分地观察 FGPC-RED 控制器的性能, 在 400 个连接时, 设置了 Q_T 为 200 与 600



(a) Q_T 为 200 的情形



(b) Q_T 为 600 的情形

图 5 400 个 TCP 连接时, Q_T 为 200 与 600 时的队列长度

Fig 5 The queue length when TCP link is 400, and Q_T is 200 and 600

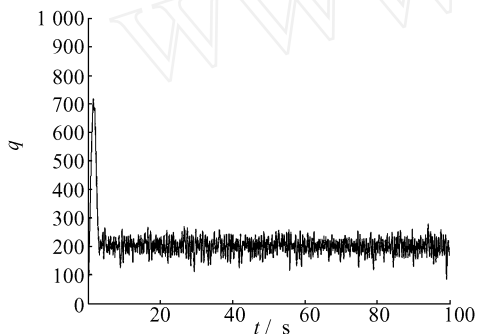
从图 5 中,可以确认 FGPC-RED 控制器可以控制目标队列长度为任何期望值.可以说,在单纯 TCP 连接时,可以成功地控制队列长度,从而可以有效地避免网络的拥塞.

4.2 TCP与 UDP并存的仿真

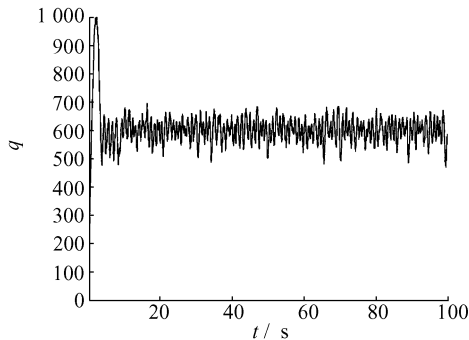
在实际的网络中,除了 TCP 连接外还有 HTTP、UDP 等其他连接方式.为了使仿真更贴近于实际情况,测试系统在多种协议并存时进行仿真.

网络的拓扑结构同图 3,仍有 400 个 TCP 连接,但同时添加了 UDP 扰动.在试验中,添加了 400 个 UDP 流,每个 UDP 流随机启动与结束,平均每个 UDP 流运行时间为 1.0 s Q_T 为 200 与 600 的情况在图 6 中进行了描述.

每个 UDP 连接参数设置如下:包的大小为 500 bit,发送速率为 50 Kbps



(a) Q_T 为 200 的情形



(b) Q_T 为 600 的情形

图 6 400 个连接时 Q_T 为 200 与 600 时的队列长度

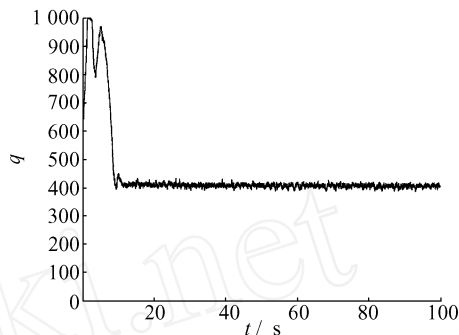
Fig 6 The queue length when link is 400 and Q_T is 200 and 600

从图 6 中,可以确认 FGPC-RED 控制器可以控

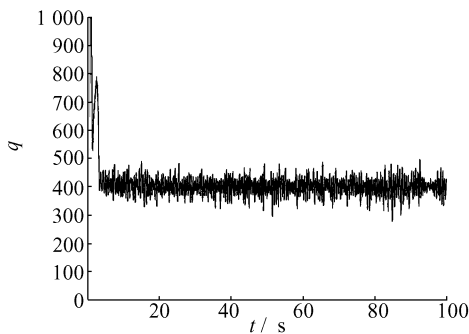
制目标队列长度为任何期望值.

4.3 基于 TCP 连接在不同传播延时下的仿真

为了测试 FGPC-RED 控制器在不同链路容量的性能,将链路的容量从 45 Mbps 变换到 15 Mbps 及 75 Mbps,其他的参数保持不变.采用 400 个节点的测试结果如图 (7) 所示,从结果可以看到系统同样稳定,并且可以得到上面同样的结论.



(a) 链路容量为 15 Mbps 的情形



(b) 链路容量为 75 Mbps 时的情形

图 7 400 个节点,链路容量为 15 Mbps 及 75 Mbps 时的队列长度

Fig 7 The queue length when nodes is 400 and the link capacity is 15 Mbps and 75 Mbps

4.4 与其他算法的比较

在验证了 FGPC-RED 控制器可以有效地控制队列长度之后,本部分主要研究的是系统的吞吐量及链路的利用率.基于 FGPC-RED 算法为 RED 的改进算法,所以与原始 RED 进行了比较,表 1 显示了在不同参数下,2 种算法性能的比较.通过比较可知采用 FGPC-RED 算法的拥塞控制算法,在减少数据包重发方面获得了较好的效果,可以减小链路的负载,间接减小链路的延迟.

表 1 GPC-RED算法性能

Table 1 The performance of GPC-RED algorithm						
节点数	期望队长	接包数	丢包数	重发数	重发率	
60	200	611 719	8 901	10 857	0.017 75	
60	400	612 073	6 473	8 252	0.013 48	
60	600	612 230	5 133	6 937	0.011 33	
100	200	642 577	16 733	21 129	0.032 88	
100	400	645 228	12 549	15 204	0.023 56	
100	600	645 578	9 935	12 378	0.019 17	
400	200	638 584	64 327	74 398	0.116 50	
400	400	638 849	55 312	65 294	0.102 21	
400	600	639 644	46 539	55 928	0.087 44	

注:接收总数为接收端接收到的有效包数,丢包数为路由器所丢弃的包数,重发数为发送端由于延时或丢包重传的包数.

表 2 RED算法性能

Table 2 The performance of RED algorithm						
节点数	q_{min}	q_{max}	接包数	丢包数	重发数	重发率
60	50	200	640 506	9 995	14 040	0.021 92
60	200	400	641 782	7 878	11 740	0.018 29
60	400	600	641 954	5 429	10 180	0.015 86
100	50	200	633 112	18 921	28 014	0.044 25
100	200	400	640 973	13 518	18 926	0.029 53
100	400	600	641 772	9 457	12 378	0.019 29
400	50	200	625 885	60 615	82 764	0.132 24
400	200	400	633 487	58 843	73 415	0.115 89
400	400	600	634 719	48 550	63 544	0.100 11

5 结束语

由于互联网的飞速发展和业务的多样化,TCP/IP拥塞控制对互联网鲁棒性越来越重要.端到端的TCP拥塞控制机制由于其局限性,难以适应形势的发展.因此,路由器必须主动地参与到网络拥塞控制中去,和TCP拥塞控制机制相结合,解决网络的拥塞控制问题,确保互联网的鲁棒性. AQM技术是路由器进行拥塞控制的关键技术.和传统的“去尾”等PQM技术相比,AQM通过有效地管理队列长度,减

小了丢包率、降低了端到端延迟、提高了吞吐量、避免了“死锁”现象的发生.因此,AQM已经成为目前的研究热点之一.本文从控制的角度将FGPC控制算法应用到路由队列管理上,取得较好的效果.通过在NS2的仿真,显示了采用传统控制算法对RED系统模型辨识的可行性及FGPC对队列管理的潜在应用价值.仿真表明,采用FGPC的RED算法能有效地控制路由队列的长度,从而有效地避免拥塞现象的发生,对提高网络吞吐率、提供可预测的链路延迟及获得较小的丢包率具有实在的价值.

参考文献:

[1] POSTEL J. Transmission control protocol [R]. IETF RFC793, 1981.

[2] YEOM I, REDDY A L N. Modeling TCP behavior in a differentiated services network [R]. Texas A&M Technical Report, 1999.

[3] FLOYD S, FALL K. Router mechanisms to support end-to-end congestion control [R]. LBL Technical Report, 1997.

[4] BRADEN B. Recommendations on queue management and congestion avoidance in internet [R]. RFC 2309, 1998.

[5] NORMEY-RICO J E, CAMCHO E F. Robust design of GPC for process with time delay [J]. J Robust Nonlinear Control, 2000, 10 (7): 1105-1127.

[6] 赵怀彬,王诗宓. 快速递推广义预测控制 [J]. 控制理论与应用, 1998, 15 (2): 190-196.

ZHAO Huaibin, WANG Shifu. Fast recursive generalized predictive control [J]. Control Theory and Applications, 1998, 15 (2): 190-196.

[7] FLOYD S, JACOBSON V. Random early detection gateways for congestion avoidance [J]. IEEE/ACM Transaction Networking, 1993, 1 (4): 397-413.

[8] RICHalet J. Industrial applications of model-based predictive control [J]. Automatica, 1993, 29 (10): 1251-1274.

[9] 舒迪前. 预测控制系统及其应用 [M]. 北京:机械工业出版社, 1996.

[10] 席裕庚. 预测控制 [M]. 北京:国防工业出版社, 1995.

[11] 王秀峰. 系统建模与辨识 [M]. 北京:电子工业出版社, 2004.

[12] SANDEEP B. Improving simulation for network research 99-702b [R]. Los Angeles: University of Southern Califor-

nia, 1999.

作者简介:



陈增强,男,1964年生,教授,博士生导师,中国系统仿真学会理事、中国自动化学会控制理论专业委员会委员、中国自动化学会过程控制专业委员会委员、中国优选法统筹法与经济数学研究会复杂系统专业委员会理事,主要研究方向为智能控制、复杂网络控制等.获得省部级科技进步奖 4 次,发表学术论文 80 余篇.



刘忠信,男,1975年生,副教授,博士,主要研究方向为计算机网络、复杂网络控制等.获得省部级科技进步奖 1 项次,发表学术论文 30 余篇.



袁著祉,男,1937年生,教授,博士生导师,《自动化学报》《控制理论与应用》杂志编委,IEEE 高级会员,主要研究方向为智能控制、预测控制等.获得教育部科技进步一等奖和国防科工委光华一等奖,发表学术论文 150 余篇.

The 3rd International Symposium on Intelligence Computation and Applications (ISICA 2008)

第三届智能计算及其应用国际会议 (ISICA 2008)

The 3rd International Symposium on Intelligence Computation and Applications (ISICA 2008) will be held on December 19 - 21, 2008 in Wuhan , China . Following the successful ISICA 2005 and ISICA 2007 sponsored by the China University of Geosciences (CUG) , one proceedings of ISICA 2007 published by Springer has been indexed by EI and ISTP, and the other proceedings of ISICA 2007 published by Press of China University of Geosciences has been indexed by ISTP.

Topics:

The emphasis of ISICA 2008 will be on the development of theories and methodologies in the field of computational intelligence and their applications . It covers all topics in intelligent computation, including, but not limited to:

- Evolutionary computation
- Evolutionary data mining
- Neural networks
- Evolutionary learning systems
- Fuzzy systems
- Evolutionary multi - objective and dynamic optimization
- Ant colony optimization
- Molecular & quantum computing
- Artificial life
- Representations & operators
- Bioinformatics & bioengineering
- Real - world applications
- Combinatorial & numerical optimization
- Theory of intelligent computation

Important Due Dates:

Paper Submission: July 15, 2008

Decision Notification: September 1, 2008

Camera - Ready Submission: September 15, 2008

Website: <http://isica.cug.edu.cn>