第34卷 第21期 计 算 机 工 程 2008年11月
Vol.34 No.21 Computer Engineering November 2008
·网络与通信· 文章编号：1000—3428(2008)21—0086—03 文献标识码：A 中图分类号：TP393

Ad Hoc 网络 PI 主动队列稳定区域研究

陈 亮[1,2]，张 宏[1]，胡为民[2]
(1. 南京理工大学计算机学院，南京 210094；2. 南通纺织职业技术学院信息系，南通 226007)

摘 要：主动队列管理(AQM)的比例积分(PI)算法可以有效控制 Ad hoc 网络的瓶颈节点队列长度，其稳定性是实现拥塞控制的基础。针对目前 PI-AQM 设计大多缺乏稳定区域的理论分析问题，该文根据 Ad hoc 网络的多跳、时延特点，分析 PI 算法在时延无线网络中的稳定性，给出无时延和大时延下 PI 算法的稳定区域，以便进一步优化设计 PI 控制器。通过 Matlab 和 NS2 仿真验证了稳定区域结论的正确性。
关键词：Ad hoc 网络；时延；比例积分；稳定区域

Research on Stable Region of PI Active Queue Management in Ad Hoc Network

CHEN Liang[1,2], ZHANG Hong[1], HU Wei-min[2]
(1. School of Computer, Nanjing University of Science and Technology, Nanjing 210094;
2. Information Department, Nantong Textile Vocational Technology College, Nantong 226007)

【Abstract】 Proportional Integral(PI) algorithm in Active Queue Management(AQM) can control the queue size of bottleneck nodes in Ad hoc network effectively, and stability of the algorithm is the base of congestion control. Now the designs of PI-AQM always lack theoretic analysis of stable region. This paper analyzes stability characteristic of PI-AQM in wireless and delay Ad hoc network because of the multi-hop and large-delay, and gives stable regions of PI-AQM controllers in delay and non-delay networks to improve PI controllers. Matlab and NS2 simulations validate the conclusion.
【Key words】 Ad hoc network; delay; Proportional Integral(PI); stable region

1 概述

Ad hoc网络是一种特殊的自组织对等式多跳移动通信网络，是由一组带有无线通信收发装置的移动终端节点组成的一个多跳的临时性无中心网络[1]。Ad hoc网络的资源非常有限，而且随着节点的增多以及TCP流发送端和接收端跨度的增加，网络有效带宽将变得很小，因此，拥塞研究显得十分重要。但是，基于有线的拥塞控制方案并不完全适用于Ad hoc网络。

主动队列管理(Active Queue Management, AQM)机制通过动态管理并检测路由中的数据堆栈长度来预知可能产生的网络拥塞，并及时通知源端，使之能够及早采取措施，从而避免更严重的数据丢失[2]。具有路由器功能的中间节点是Ad hoc网络的重要组成部分，也是网络拥塞状态最直接的感受者，在这些中间节点中引入相应的拥塞控制机制，使网络本身参与资源的控制工作，可以有效实现对拥塞的监测和预防。由于主动队列管理在无线网络中的应用研究尚未充分展开，因此很有必要对Ad hoc网络中基于主动队列管理的拥塞控制进行研究。

基于控制理论的 AQM 具有稳定性和鲁棒性，但也存在一些问题，如控制器设计大多属于工程试凑，参数选择具有一定随意性，缺乏对控制的稳定性原理分析，一般没有给出控制器参数稳定区域集合。为了更好地设计 AQM 控制器，本文在 Ad hoc 网络环境下，针对主动队列管理的比例积分(Proportional Integral, PI)算法，详细分析了时延系统的稳定性，给出了 PI 控制器参数的稳定区域集合。通过对得到的稳定区域进行 Matlab 和 NS2 仿真，验证了方法的有效性。

2 PI 算法简介

自从 1998 年Braden等人提出主动队列管理的概念后，一系列主动队列管理算法被提出。PI控制作为最早提出的基于控制理论的AQM算法，其优点是在一定程度上克服了连接数目N和往返传输时间 R_0 等扰动，把队列长度控制在一个目标值 q_0 附近，仿真结果表明，PI算法能克服稳态误差，收敛速度也比RED算法快。由于具有鲁棒性和降低稳态误差的特点，PI控制成为AQM控制器较理想的方案[3]。但通过工程设计发现，它的性能表现和控制器比例、积分系数的设置关系密切，具体表现在参数的确定上。PI控制大多采用试凑法，优化设计也往往根据过去的经验，缺乏稳定性方面的理论依据，从而导致系统的动态和稳态性能得不到可靠保障。

对于控制对象的时延环节，如果时延较小，工程中常用的方法是忽略其时延项，近似看作非时延对象，但对于Ad hoc网络而言，无线多跳网络的特点决定其传输时延往往不能被忽略，而在大时延网络中，所有典型算法控制的队列均出现了大幅度振荡。队列的大幅度振荡一方面增加了端到端的时延抖动，同时由于空队列出现概率的加大，链路利用率必然会降低，这 2 点都违背了最初提出的AQM设计目标[4]。因此，必须分析大时延网络控制系统的稳定性。本文通过比较分析

基金项目：江苏省自然科学基金资助项目“面向安全的移动自组网络自适应适配构架”(BK2007593)
作者简介：陈 亮(1982－)，男，博士研究生，主研方向：无线自组网，网络拥塞控制；张 宏，教授、博士生导师；胡为民，副教授
收稿日期：2007-12-10 **E-mail：**njustchenliang@yahoo.com.cn

TCP/AQM非时延对象和时延对象的稳定性，得到时延对象的稳定特性，以便进一步设计优选PI控制器参数，达到更好的PI控制效果。

3 PI 算法稳定区域分析

3.1 控制模型

基于流体流理论，Misra等把Internet中的信息流看作一个连续的流体，给出了TCP/AQM 的微分方程，从而可以使用控制理论研究AQM。Hollot等人又对该微分方程做了线性化处理，建立了AQM作用下TCP连接的拥塞窗口的非线性动态模型[5]。Misra提出的Internet非线性动态模型如下:

$$\begin{cases}\dot{W}(t)=\dfrac{1}{R(t)}-\dfrac{W(t)}{2}\dfrac{W(t-R(t))}{R(t-R(t))}p(t-R(t))\\ \dot{q}(t)=-C+\dfrac{N(t)}{R(t)}W(t)\end{cases} \tag{1}$$

其中，$\dot{W}(t)=\dfrac{\mathrm{d}W(t)}{\mathrm{d}t}$，$\dot{q}(t)=\dfrac{\mathrm{d}q(t)}{\mathrm{d}t}$ ；W 为 TCP 拥塞窗口大小(单位：分组)；q 为队列长度(单位：分组)；R 为往返时间(单位：s)；C 为链路容量(单位：分组/s)；N 为负载(TCP 连接数量)；$p(t)$ 为分组丢弃概率。设系统稳态下的往返传输时间是 R_0，TCP 连接数目是 N，理想队列长度是 q_0，丢包率是 p_0，并认为研究对象为具有延时的网络，经过线性化和拉普拉斯变换，得到受控对象的传递函数：

$$P(s)=\frac{(R_0C)^3/(2N)^2}{(R_0^2Cs/(2N)+1)(R_0s+1)}\mathrm{e}^{-R_0s}=\frac{L}{(s+a)(s+b)}\mathrm{e}^{-R_0s} \tag{2}$$

其中，$L=C^2/(2N)$ ；$a=2N/(CR_0^2)$ ；$b=1/R_0$ 。

令 $P_0(s)=\dfrac{L}{(s+a)(s+b)}$ 为被控对象非时延部分。线性化后的 TCP/AQM 结构如图 1 所示，其中，$C(s)$ 是控制器；$P_0(s)\mathrm{e}^{-sR_0}$ 是被控对象 TCP/AQM 系统。

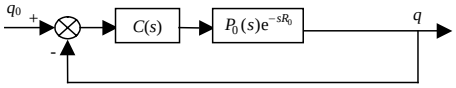

图 1 控制系统框图

根据控制理论设计 PI 控制器，其传递函数为

$$C(s)=k_p+k_i/s \tag{3}$$

参数设置为：$q_0=20$ packet, $R_0=0.25$ s, N=10, C=1 Mb/s=250 packets/s。

3.2 非时延系统的稳定性分析

为了简化问题及对比，先对系统中的非时延部分 $P_0(s)$ 单独进行稳定性分析。

非时延对象闭环控制系统的特征方程为

$$1+P_0(s)C(s)=0$$

化简得

$$s^3+(a+b)s^2+(ab+Lk_p)s+Lk_i=0 \tag{4}$$

将式(4)的多项式系数排成如下的劳斯阵列：

s^3	1	$ab+Lk_p$
s^2	$(a+b)$	Lk_i
s^1	$[(a+b)(ab+Lk_p)-Lk_i]/(a+b)$	0
s^0	Lk_i	

根据劳斯稳定判据：劳斯阵列的第 1 列所有项均为正号，系统稳定。因为 Ad hoc 网络的性能参数(C, N, R_0)不会为负，显然，$(a+b)>0$；同时仅考虑正增益，可得 $Lk_i>0$。即只要 $[(a+b)(ab+Lk_p)-Lk_i]/(a+b)>0$，系统稳定。化简得非时延控制系统的稳定区域为 $0<k_i<(a+b)(ab+Lk_p)/L$，如图 2 所示。可以看出，图 2 中的稳定分界线即 $k_i=(a+b)(ab+Lk_p)/L$。

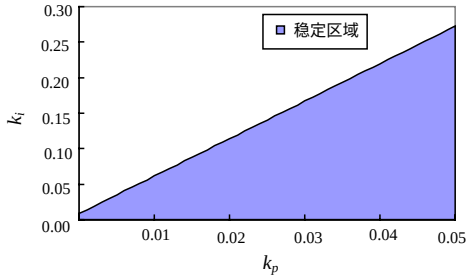

图 2 无时延系统稳定区域

3.3 时延系统的稳定性分析

从式(2)来看，系统中时延的加入使得经典控制理论的稳定区域分析不再适用。对于含时延的控制系统，其闭环特征方程为 $1+P_0(s)C(s)\mathrm{e}^{-sR_0}=0$。化简得

$$s^3+(a+b)s^2+abs+L(k_ps+k_i)\mathrm{e}^{-sR_0}=0 \tag{5}$$

令 $s=\mathrm{j}\omega$，同时根据欧拉方程

$$\mathrm{e}^{-\mathrm{j}\omega R_0}=\cos(\omega R_0)-\mathrm{j}\sin(\omega R_0)$$

其中，$\omega\in(0,+\infty)$，式(5)化为

$$\{-(a+b)\omega^2+L[k_p\omega\sin(\omega R_0)+k_i\cos(\omega R_0)]\}+\mathrm{j}\{(ab\omega-\omega^2)+L[k_p\omega\cos(\omega R_0)-k_i\sin(\omega R_0)]\}=0$$

复数为 0，则其实部、虚部都为 0，可得

$$\begin{cases}k_p\omega\sin(\omega R_0)+k_i\cos(\omega R_0)=(a+b)\omega^2/L\\ k_p\omega\cos(\omega R_0)-k_i\sin(\omega R_0)=(\omega^3-ab\omega)/L\end{cases}$$

联立上述两式解得

$$\begin{cases}k_p=\dfrac{(a+b)\omega\sin(\omega R_0)+(\omega^3-ab)\cos(\omega R_0)}{L}\\ k_i=\dfrac{(a+b)\omega^2\cos(\omega R_0)-\omega(\omega^2-ab)\sin(\omega R_0)}{L}\end{cases}$$

根据上式，可在 k_p-k_i 平面上绘出时延系统的稳定区域，如图 3 所示。可以发现，时延系统稳定区域在非时延系统稳定区域的内部，即它是非时延系统稳定区域的子集。

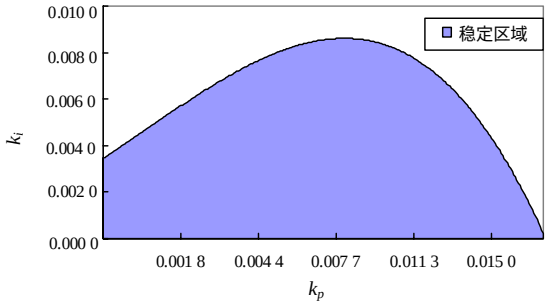

图 3 时延系统稳定区域

4 仿真

4.1 Matlab 仿真

针对时延系统，分别取图 3 所示的点(k_p,k_i)=(0.001 8, 0.002 0)、稳定区域外的点(k_p,k_i)=(0.001 8, 0.010 0)，利用 Matlab 进行仿真。分别绘制时域下阶跃响应曲线，如图 4 所示。可以看出，选择稳定区域内的点设计的 PI 控制器能很好地稳定 TCP/AQM 时延系统，而稳定区域外的点会导致时延系统震荡剧烈并最终发散。

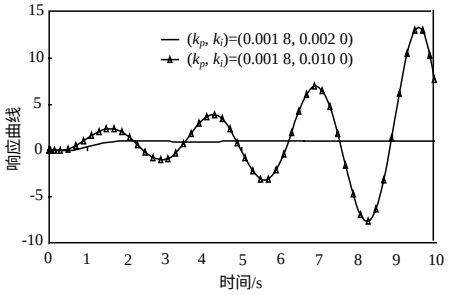

图 4　时延系统的阶跃响应曲线

4.2　NS2 仿真

本文的网络仿真环境为Windows 2000 +cygwin+NS2.27，NS2 是一种面向对象的网络仿真器，它本身含有一个虚拟时钟，采用离散事件驱动作为引擎。它的底层仿真引擎主要由C++编写，同时利用面向对象的工具命令语言(OTCL)作为仿真时的命令和配置语言[6]。

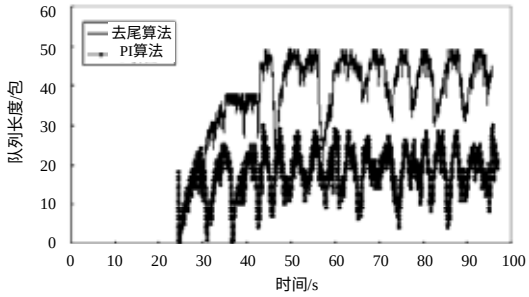

图 5　去尾算法和 PI 算法的队列长度曲线

无线环境配置为：在 500 m×400 m 的平面上，无线节点 S_1, S_2, S_3,…, S_N 通过瓶颈节点 *R* 向节点 *D* 发送数据，其应用层上运行 FTP 服务。无线节点 *R* 为瓶颈节点，其接口队列长度为 50 个包，*N*=10，具体参数配置同 3.1 节。*R* 的接口队列采用 PI 算法，令(k_p, k_i)=(0.001 8, 0.002 0)，仿真时间为 100 s。如图 5 所示，与传统的去尾算法相比，PI 控制器取得了较好的控制效果，队列长度维持在理想值 q_0 = 20 附近。

5　结束语

分析 PI 主动队列管理算法的稳定区域是设计优化 PI 控制器的基础。本文分析了在 Ad hoc 网络环境下带时延环节和无时延环节的 TCP/AQM 被控对象，通过在 $k_p - k_i$ 平面上绘制 PI 控制器的稳定边界线，分别得出了两者在 $k_p - k_i$ 平面上的 PI 算法稳定区域，为 TCP/AQM 控制系统的设计提供了有效的方法。Matlab 和 NS2 仿真表明，稳定区域分析可以有效辅助 PI 控制器的设计，为进一步设计适合 Ad hoc 网络的 AQM 算法提供了稳定性方面的依据。

参考文献

[1] Conti M, Giordano S. Special Issue on Mobile Ad Hoc Network[J]. Cluster Computing, 2002, 5(2): 117-131.

[2] Hollot C, Misra V, Towsley D. On Designing Improved Controllers for AQM Routers Supporting TCP Flows[C]//Proceedings of the INFOCOM'01. Alaska, USA: IEEE Computer Society, 2001.

[3] Wu Wei, Ren Yong, Shan Xiuming. A Self-configuring PI Controller for Active Queue Management[C]//Proc. of Asia-Pacific Conference on Communications. Tokyo, Japan: [s. n.], 2001.

[4] Braden B, Clark D, Crowcroft J, et al. Recommendations on Queue Management and Congestion Avoidance in the Internet[S]. RFC 2309, 1998.

[5] Morari V, Gong W, Towslay D. Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows[C]//Proc. of ACM SIGCOMM'00. [S. l.]: ACM Press, 2000.

[6] 杨吉文, 张卫东. 基于 NS2 的主动队列管理仿真研究[J]. 计算机工程, 2006, 32(17): 189-191.

（上接第 85 页）

下面通过实验比较在没有使用 BCLFL 策略与使用该策略的服务器性能，这里采用 Banga 和 Drushel 在文献[4]中提出的一种用双进程方法来模拟同一时间内产生较大数量客户端并发地向服务器发出请求的测试用例。利用文献[4]中双线程测试法对原来的 CMS 系统和采用 BCLFL 策略优化的系统分别进行测试对比，结果如表 2 所示。

表 2　BCLFL 策略测试　　ms

测试项目	测试内容	CMS	CMS(BCLFL)
	Hit Count	365 838	365 834
Time to first Byte	Average	13.99	2.43
	Min	3.98	2.26
	25%	12.36	2.64
	50%	12.53	2.74
	75%	14.39	2.81
	Max	271.02	43.35
Time to last Byte	Average	16.10	3.30
	Min	3.99	2.84
	25%	14.53	3.23
	50%	14.72	3.33
	75%	16.81	3.42
	Max	289.75	44.02

表 2 的实验结果显示 CMS 系统在使用了 BCLFL 策略之后对于响应时间的提高十分明显，平均响应时间比原系统降低了约 80%。

3　结束语

在利用 Apache, PHP 和 MySQL 构建的 Web 服务器中，文中利用 PHP 的各种特性函数，在充分考虑文件大小、访问频率以及链接的方式等因素的基础上，使用 BCLFL 策略进行优化。实验表明，该策略可以明显改善服务器的性能。

参考文献

[1] Abrams M, Standridge C, Abdulla G, et al. Caching Proxies: Limitations and Potentials[C]//Proc. of the 4th International World Wide Web Conference. Boston, USA: [s. n.], 1995.

[2] Fuecks H. The PHP Anthology: Object Oriented PHP Solutions[EB/OL]. (2003-12-01). http://www.ebookee.com.cn/The-PHP-Anthology-Object-Oriented-PHP-Solution-Volume-1_105586.html.

[3] Shim J, Scheuermann P, Vingralek R. Proxy Cache Algorithms: Design, Implementation and Performance[J]. IEEE Trans. on Knowledge and Data Engineering, 1999, 11(4): 549-562.

[4] Banga G, Druschel P. Measuring the Capacity of a Web Server[C]//Proc. of USENIX Symposium on Internet Technologies and Systems. Monterey, California, USA: [s. n.], 1997.