

动态自适应的Internet拥塞控制系统的PID调解器

杨洪勇¹, 王福生², 段惠东²

(1. 鲁东大学计算机学院, 烟台 264025; 2. 鲁东大学情报咨询部, 烟台 264025)

摘要: Internet网络拥塞控制算法决定着网络的服务质量。把Internet通信系统看作具有通信时延的闭环反馈系统, 建立了一个PID调解器控制Internet连接节点的拥塞率。采用特定相位裕量方法动态调整PID控制器的参数, 使控制器能在线自适应网络系统中的变化。最后仿真验证了新的主动队列管理策略的有效性和鲁棒性。

关键词: Internet拥塞控制; 主动队列管理策略; PID调解器

Dynamical Adaptive PID Controller of Internet Congestion Control System

YANG Hongyong¹, WANG Fusheng², DUAN Huidong²

(1. Department of Computer Science and Technology, Ludong University, Yantai 264025;

2. Information Consultation Office, Ludong University, Yantai 264025)

【Abstract】 Internet congestion control algorithms decide the quality of service (QoS). By considering Internet system as a closed-loop feedback system with time-delay, a PID controller is established as active queue management (AQM) scheme to adjust the congestion marked possibility. The controller parameters are dynamically tuned by applying especial reference phase method for adapting the change of the Internet system. Finally, computer simulations show the validity and robustness of the new AQM scheme.

【Key words】 Internet congestion control; Active queue management (AQM) scheme; PID controller

现在Internet的拥塞控制是基于Jacobon提出的TCP流量控制算法^[1]改进得到的TCP Reno。由于只依靠终端系统的TCP流量控制很难保证网络信息的可靠传输, 中间节点的主动队列管理(AQM)策略采用由RFC2309推荐的RED拥塞控制算法^[2], 来提高网络的服务质量。但是近期的大量研究表明RED算法对网络的参数设置和运行状况比较敏感, 会出现节点队列大幅振荡、吞吐量降低、时延增加等网络不稳定现象, 对网络的服务质量影响很大。对AQM策略的进一步研究相继产生了许多改进算法: SRED、ARED、REM等, 这些策略减少了RED算法在连接节点的队列振荡, 对原来RED算法的拥塞控制策略和算法参数进行了一定程度的改进。

本文从控制理论的角度出发, 提出一个在线调整的PID控制器作为新的AQM策略, 来提高Internet数据传输能力。新的AQM策略和RED算法相比, 对网络负载干扰和噪声干扰具有较强的鲁棒性, 能很好地把连接节点的队列控制到期望的队列长度。

1 具有特定相位裕量的PID自校正控制系统

自校正控制系统是一类主要的自适应控制系统。因为其原理简单、容易实现, 现已经广泛地用在参数变化、有时延和时变过程, 以及具有随机扰动的复杂系统中。在工业过程控制中, 常规的PID调节器由于具有原理简单、整定方便、鲁棒性强和不需要对象的精确模型的优点, 得到广泛的应用。现在提出的PID自调整方法很多, 如Ziegler-Nichols提出的频率响应法、Astrom-Hagglund提出的相增益法、误差积分性能准则等。本文采用Astrom-Hagglund提出的相增益法建立一个在线自整定PID控制器来调整网络系统的拥塞率, 从而达到控制网络拥塞, 提高网络服务能力的目的。

假设系统被控对象 $G_0(s) = \frac{k_0 e^{-s}}{Ts+1}$, PID控制器传递函数为

$$G_c(s) = K_p(1 + \frac{1}{sT_i} + sT_d)$$

由PID调节器构成的控制系统(见图1)的开环传递函数为

$$G_k(s) = K_p(1 + \frac{1}{sT_i} + sT_d)G_0(s) \tag{1}$$

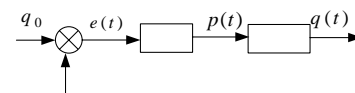


图1 由PID调节器构成的闭环控制系统

假定 $G_0(s)$ 的Nyquist曲线和负实轴交点C的频率是 ω_c , 频率 ω_c 称为临界频率, 相应的临界周期 $t_c = 2\pi / \omega_c$, 而临界增益 K_c 和对象频率特性在该点的幅值 $|G_0(j\omega_c)|$ 有下列关系: $1/K_c = |G_0(j\omega_c)|$ 。如果令闭环系统的相位裕量为 φ_m , 则根据相位裕量的定义, 系统开环频率特性在 ω_c 时的幅角应为

$$\arg(1 + j\omega_c T_d + \frac{1}{j\omega_c T_i}) - \pi = -\pi + \varphi_m$$

可得

$$\omega_c T_d - \frac{1}{\omega_c T_i} = \tan \varphi_m$$

在整定PID参数量时, 一般取 $T_i = 4T_d$ 。

基金项目: 国家自然科学基金资助项目(60574007); 鲁东大学校基金资助项目(22320301)

作者简介: 杨洪勇(1967-), 男, 博士后、教授, 主研方向: Internet拥塞控制, 鲁棒控制, 复杂网络; 王福生、段惠东, 学士、工程师

收稿日期: 2006-03-22 **E-mail:** hyyang@yeah.net

代入上式可以得到关于 T_d 的二阶方程，其解

$$T_d = \frac{\tan \varphi_m + \sqrt{1 + \tan^2 \varphi_m}}{2\omega_c} \quad (2)$$

进而根据相位裕量的定义可知，系统开环频率特性在相位为 φ_m 处的幅值为 1，即

$$K_p \sqrt{1 + (\omega_c T_d - \frac{1}{\omega_c T_i})^2} |G_0(j\omega_c)| = 1$$

进一步简化为

$$K_p = \frac{\cos \varphi_m}{|G_0(j\omega_c)|} = K_c \cos \varphi_m \quad (3)$$

由式(2)、式(3)可以求得对应于某一 φ_m 值的 PID 整定参数，如表 1 所列。

表 1 利用相位裕量设计 PID 参数

φ_m	K_p	T_i	T_d
30°	$0.87K_c$	$0.55t_c$	$0.14t_c$
45°	$0.71K_c$	$0.77t_c$	$0.20t_c$
60°	$0.50K_c$	$1.29t_c$	$0.30t_c$

2 Internet 拥塞控制系统的自调整 PID 控制器

2.1 Internet 拥塞控制算法的非线性动态模型

Misra 等^[3]把Internet中的信息流看作一个连续的流体，建立了AQM作用下TCP连接的拥塞窗口的非线性动态模型。Hollot 等人^[4]以该动态模型为基础，研究了Internet的主动队列管理策略RED算法的动态特性。Misra提出的Internet非线性动态模型如下：

$$\begin{aligned} \dot{W}(t) &= \frac{1}{D(t)} - \frac{W(t)W(t-D(t))}{2D(t)} p(t-D(t)) \\ \dot{q}(t) &= \frac{N(t)}{D(t)} W(t) - C(t) \end{aligned} \quad (4)$$

其中 $W(t)$ 为窗口大小， $N(t)$ 为 TCP 连接数， $C(t)$ 为连接节点的服务能力， $D(t)$ 为往返时延， $q(t)$ 为连接节点平均队列长度， $q(t)$ 为连接节点的拥塞率。假设系统的往返时延为 $D(t) = D_0$ ，同时假设连接节点的服务能力 $C(t) = C$ ，系统的 TCP 连接数为 N ，则系统的平衡点 (W_0, p_0) 满足：

$$W_0 = \frac{D_0 C}{N}, \quad p_0 = \frac{2}{W_0^2}$$

在系统平衡时，系统中连接节点中的队列长度也达到平衡。在平衡点处对系统进行线性化，忽略发送窗口的时延动态，线性化的系统动态方程为

$$\begin{aligned} \delta \dot{W}(t) &= -\frac{2N}{D_0^2 C} \delta W(t) - \frac{D_0 C^2}{2N^2} \delta p(t-D_0) \\ \delta \dot{q}(t) &= \frac{N}{D_0} \delta W(t) - \frac{1}{D_0} \delta q(t) \end{aligned} \quad (5)$$

其中 δW 、 δq 、 δp 分别为系统动态 $W(t)$ 、 $q(t)$ 、 $p(t)$ 在平衡点的扰动， D_0 为平衡往返时延。对上式进行 Laplace 变换得到时滞非线性动态模型的传递函数 $G_1(s)$ ，与 RED 主动队列管理策略构成闭环系统，其中

$$G_1(s) = \frac{K_1 e^{-sD_0}}{(T_1 s + 1)(T_2 s + 1)}, \quad K_1 = \frac{D_0^3 C^3}{4N^2}, \quad T_1 = D_0, \quad T_2 = \frac{D_0^2 C}{2N}$$

2.2 自调整 PID 控制器

网络拥塞控制系统的被控对象传递函数为

$$G_1(s) = \frac{K_1 e^{-sD_0}}{(T_1 s + 1)(T_2 s + 1)}$$

用一阶时滞模型

$$G_0(s) = \frac{k_0 e^{-\tau s}}{T s + 1}$$

来拟合，可得

$$k_0 = K_1, \quad \tau = T_1 + T_2 + D_0 - \sqrt{T_1^2 + T_2^2}, \quad T = \sqrt{T_1^2 + T_2^2}$$

在网络模型中用自调整PID控制器代替RED算法，根据不同的网络参数设置，动态调整PID控制器的参数 K_p, T_i, T_d 。

在实际应用中，对于离散采样系统 PID 的计算机实现也采用离散化形式，用数字形式的差分方程代替连续时间系统的微分方程。积分项和微分项用求和及增量式表示，得到下面的离散的 PID 表达式：

$$p(n) = k_p (e(n) + \frac{\Delta T}{T_i} \sum_{j=0}^n e(j) + \frac{T_d}{\Delta T} (e(n) - e(n-1)))$$

其中 ΔT 为系统的采样周期。整理可得

$$\Delta p(n) = E_0 e(n) + E_1 e(n-1) + E_2 e(n-2) \quad (6)$$

其中

$$E_0 = k_p (1 + \frac{\Delta T}{T_i} + \frac{T_d}{\Delta T}), \quad E_1 = -k_p (1 + \frac{2T_d}{\Delta T}), \quad E_2 = k_p \frac{T_d}{\Delta T}, \quad k_i = k_p \frac{\Delta T}{T_i}, \quad k_d = k_p \frac{T_d}{\Delta T}$$

假设网络系统连接节点的服务能力为 10Mbps (1 250packets/s，分组大小 1 000b/packet)，TCP 连接数 60，传播时延为 0.4s，采样频率为 60 次/s。可以得到 PID 控制器为

$$p(n) = p(n-1) + 0.0112e(n) - 0.0224e(n-1) + 0.0109e(n-2) \quad (7)$$

一般令 $p = \min(\max(0, p(n)), 1)$ ，使得网络拥塞概率介于 0 与 1 之间。

3 系统仿真

下面用仿真来测试自调整PID控制器的性能。假设网络系统连接节点的服务能力为 1 250packets/s，TCP 连接数为 60，网络往返时延为 0.02s，PID 控制器参数在线获得。RED 参数设置 $p_{\max} = 0.2$ ，指数平均队列因子 $\alpha = 0.02$ ，队列最大阈值为期望队列长度，队列最小阈值为最大阈值的一半^[5]。下面的仿真时间都为 5s。第 1 个仿真，开始 2.5s 设置期望队列长度为 300packets，时间 2.5s 后设置期望队列长度为 100packets，比较了自调整PID控制器和RED控制策略的效果(如图 2、图 3)。可见新的 AQM 策略能快速准确地将连接节点的队列长度控制到期望值。

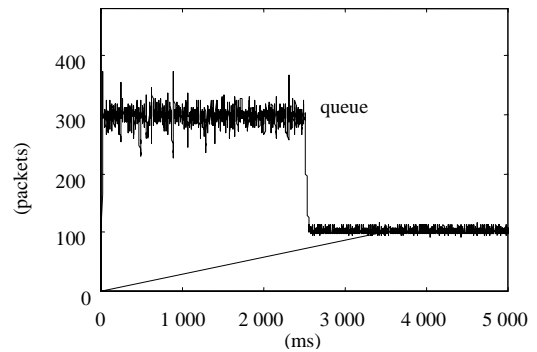


图 2 自调整 PID 控制器仿真 1

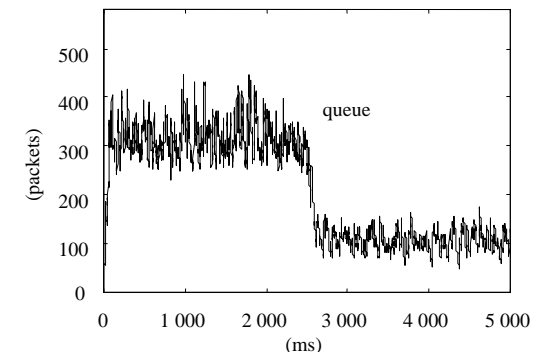


图 3 RED 算法仿真 1

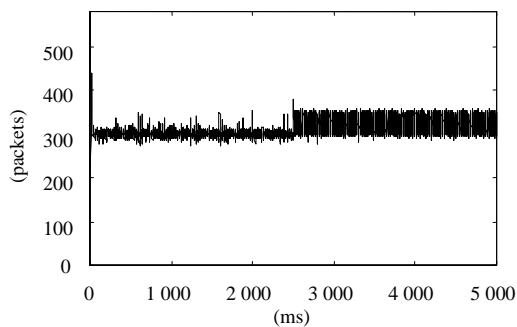


图4 自调整PID控制器仿真2

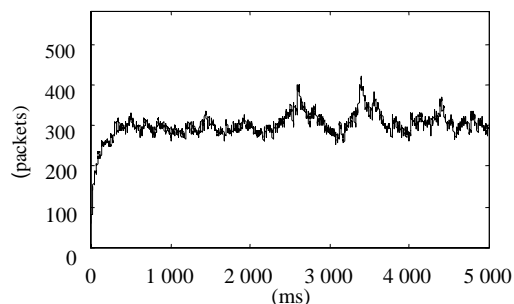


图5 RED算法仿真2

第2个仿真,假设开始2.5s与连接节点相连的TCP数为60,时间2.5s后TCP连接数变为120,其它参数同上。我们仿真了自调整PID控制器和RED控制策略的执行情况(如图

4、图5),可见新的AQM策略对负载干扰适应能力比RED的适应能力强。

4 结论

本文建立了一个自调整PID控制器来调节网络拥塞控制系统的拥塞率,控制器的参数根据网络参数的变化在线调整,保证网络系统的连接节点缓存器中队列的稳定性和网络的高吞吐量。仿真对新的AQM策略和现在使用的RED拥塞控制算法进行了比较,表明我们设计的PID控制器具有很好的适用性和很强的鲁棒性。

参考文献

- 1 Jacobson V. Congestion Avoidance and Control[C]//Proc. of SIGCOMM'88, Stanford, USA. 1988: 312-329.
- 2 Floyd S, Jacobson V. Random Early Detection Gateways for Congestion Avoidance[J]. IEEE/ACM Transactions on Networking, 1993, 1(4): 397-413.
- 3 Morari V, Gong W, Towsley D. Fluid-based Analysis of A Network of AQM Routers Supporting TCP Flows[C]//Proc. of ACM/SIGCOMM'00, Sweden. 2000: 151-160.
- 4 Hollot C, Misra V, Towsley D, et al. Analysis and Design of Controllers for AQM Routers Supporting TCP Flows[J]. IEEE Trans. on Automatic Control, 2002, 47(6): 945-959.
- 5 Floyd S. RED: Discussions of Setting Parameters[Z]. <http://www.aciri.org/floyd/>.

(上接第85页)

- $X(t) = \begin{cases} 0, & \text{时刻 } t, N1 \text{和} N3 \text{正常工作, } N2 \text{处于轻载工作状态} \\ 1, & \text{时刻 } t, N1 \text{正常工作, } N3 \text{故障, } N2 \text{接入正常工作} \\ 2, & \text{时刻 } t, N1 \text{正常工作, } N3 \text{故障, } N2 \text{也故障} \\ 3, & \text{时刻 } t, N1 \text{和} N3 \text{正常工作, } N2 \text{处于冗余故障} \\ 4, & \text{时刻 } t, N1 \text{正常工作, } N3 \text{故障, } N2 \text{处于冗余故障} \\ 5, & \text{时刻 } t, \text{三个节点都故障, 系统失效} \end{cases}$ (1)

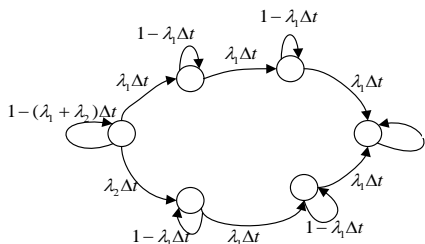


图7 系统的状态转移

列出图7模型的状态转移方程组,经过计算可求出系统处于各种状态时的概率,则系统的可靠度为

$$R(t) = P_0(t) + P_1(t) + P_2(t) + P_3(t) + P_4(t) \\ = e^{-(\lambda_1 + \lambda_2)t} + \frac{\lambda_1}{\lambda_2}(e^{-\lambda_1 t} - e^{-(\lambda_1 + \lambda_2)t}) + \frac{\lambda_1^2}{\lambda_2^2}(e^{-(\lambda_1 + \lambda_2)t} - e^{-\lambda_1 t}) \\ + \frac{\lambda_1^2}{\lambda_2}te^{-\lambda_1 t} + e^{-\lambda_1 t} - e^{-(\lambda_1 + \lambda_2)t} + \lambda_1 te^{-\lambda_1 t} + \frac{\lambda_1}{\lambda_2}(e^{-(\lambda_1 + \lambda_2)t} - e^{-\lambda_1 t})$$
 (2)

平均无故障时间为

$$MTTF = \int_0^{\infty} R(t)dt = \frac{2}{\lambda_1} + \frac{1}{\lambda_1 + \lambda_2}$$
 (3)

6 结束语

当前软件容错技术已经成为提高系统可靠性的热点问题。本文分析了容错领域的研究现状和存在的根本问题,设计了一个基于中间件的容错系统模型,提出了一种新的节点容错结构构造方法,为解决冗余、失效检测和恢复等容错的关键技术问题形成了一套较完整的解决方案,这些对将来开发分布式容错应用系统具有一定的理论和实际意义。但是其中尚有大量的问题有待解决,如应用程序合理划分为任务的方法;支持主动复制模式的容错支撑和看门狗时间的合理设置等。这些问题将在今后的研究中逐步加以解决。

参考文献

- 1 刘云龙,陈俊亮.基于数据流分析的软件容错策略[J].软件学报,1998,9(7):537-541.
- 2 Lyn M R. Software Fault Tolerance[M]. New York: Wiley & Sons Ltd.,1995.
- 3 Avizienis A. The Method of N-version Programming in Software Fault Tolerance[M]. Wiley, 1995: 23-46.
- 4 Bishop P. Software Fault Tolerance by Design Diversity in Software Fault Tolerance[M]. Wiley, 1995: 211-230.
- 5 Poledna S. Fault Tolerant Real-time System: The Problem of Replica Determinism[M]. Kluwer Academic Publisher, 1996.
- 6 Laprie J C, Arlat J, Beounes C, et al. Architectural Issues in Software Fault Tolerance[M]. Wiley, 1995: 47-80.