

主动队列管理中的 PID 控制器¹

任丰原 王福豹* 任勇 山秀明

(清华大学电子工程系 北京 100084)

*(西北工业大学计算机科学与工程系 西安 710072)

摘要 作为对终端系统上拥塞控制的一种补充,中间节点上的主动队列管理(AQM)策略在保证较高吞吐量的基础上有效地控制队列长度,从而实现了控制端到端的时延,保证 QoS 的目的。C. Hollot 等人(2001)用经典控制理论中频域校正的方法设计了用于 AQM 的 PI 控制器,但参数整定上的试凑方法不免代有盲目性;算法的瞬态性能指标也不够理想。为此,该文引入了微分环节来增强系统的响应能力,同时给出了基于稳定裕度的参数整定方法,使 PID 控制器的稳定性有了绝对保障。仿真试验表明 PID 算法的调节时间远远短于 PI 控制器,从而为在负载瞬息万变的网络环境中实现控制分组排队等待时间的目标提供了有力的技术保障。

关键词 主动队列管理, PID 控制器, 拥塞控制

中图分类号 TN919.3

1 引言

Internet 自出现以来得到了蓬勃发展,近年来更以惊人的速度增长,现有的带宽总难以完全满足用户的要求,产生网络拥塞是在所难免的,这一问题从 80 年代中期就得到了学术界和工业界的广泛重视。自 1988 年 V. Jacobson 提出了 TCP 端到端的基于窗口的拥塞控制算法^[1]以来, TCP 流量控制算法经过了 Tahoe, Reno, New Reno, SACK 等多个版本的增强与改进^[2],但所有这些工作都将研究的注意力集中到了终端系统上,通过源端对网络当前状态的感知与探测来相应调整控制发送分组的窗口大小。最近的研究表明仅仅依赖源端的流量控制很难为用户提供良好的 QoS 保证,因为它的作用是有限的,增强中间节点(路由器)的功能是一种有效的手段,显式拥塞标识(Explicit Congestion Notification, ECN)即是对网络中间节点功能的一种扩展^[3],此外,中间节点上的队列管理也严重影响着 TCP 的性能,例如,不恰当的队列管理算法可能造成 TCP 连接的全局同步、队列长时间处于满状态、处理突发业务时存在公平性问题^[4]。1998 年, B. Braden 等人在 IETF 提出了主动队列管理(Active Queue Management, AQM)的研究动议^[5],作为端到端拥塞控制的一种技术手段,期望 AQM 在减小排队时延的同时保证较高的吞吐量。具体分析, AQM 解决的问题主要包括以下 4 方面:

(1) 早期探测路由器可能发生的拥塞,并通过随机丢弃或标记分组来通知源端采取措施避免可能发生的拥塞。

(2) 公平地处理包括突发性、持久性和间隙性的各种 TCP 业务流。

(3) 避免多个 TCP 连接由于队列溢出而造成同步进入“慢启动”状态。

(4) 维持较小的队列长度,在高吞吐量和低时延之间做出合理平衡。

S. Floyd 等人提出的随机早期检测(Random Early Detection, RED)算法^[6]符合 AQM 的主旨, RFC2309 推荐它为主动队列管理的唯一候选算法。虽然 RED 已被广泛用在网络队列管理中来提高系统的综合性能,然而, RED 算法本身依旧存在着许多不完善之处。已有的研究表明^[7]: 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; RED 算法的公平性^[8]和稳定性^[9]也存在问题。为此,围绕 RED 算法本身展开了许多研究工作;有从理论角度分析 RED 算法缺陷的^[9,10];有从仿真或实验入手验证 RED 算法有效性的^[8];大多数工作试图完善和改进 RED 存

¹ 2001-07-12 收到, 2002-04-08 改回
国家自然科学基金(No.69972040)资助

在的缺陷, 于是出现了不少 RED 的派生算法, 较有影响力的有 gentle-RED^[11], WRED^[12], BLUE^[13], Stabilize-RED^[14], FRED^[15], Self-configuration RED^[16] 和 Balanced-RED^[17]. 它们中的大多数研究工作在很大程度上是依赖于直觉的、启发性的、针对局部个别问题的, 没有系统的理论作为分析和设计的依据. C.Hollot 等人用小信号理论对 V. Misra 给出的 TCP 流量控制模型^[18] 进行线性化处理后, 用经典线性控制理论分析了 RED 的稳定性^[19], 最后在文献 [20] 中给出了一种用于主动队列管理的 PI(Proportional Integral) 控制器, 增强了对队列长度的控制能力, 更有助于 AQM 技术目标的实现. 文献 [20] 的最大贡献在于将 AQM 算法的设计问题转化为控制理论中典型的调节器设计问题, 但因在时变的网络环境中很难确切地得到系统的临界放大倍数和临界振荡周期, 也就无法使用常用的 Ziegler-Nichols 方法来整定 PI 控制器的参数, 因此被迫采用了试凑方法, 虽然给出的 PI 调节器能够稳定有效的工作, 但我们有理由怀疑它的性能. 分析文献给出的仿真结果, 不难发现 PI 控制器的调节时间过长, 对路由器缓存大小的依赖过强, 最直接的原因可能是调节器缺少反映队列变化速率的微分环节造成的. 基于以上几个方面的分析, 在本研究中, 我们来设计用于主动队列管理的 PID 控制器, 期望它能够有优于文献 [20] 给出的 PI 控制器的性能. 因客观条件限制 Ziegler-Nichols 方法无法用在 PID 参数的整定中, 我们给出一种基于稳定裕度的参数整定方法, 在理论上, 应该较试凑法有效且可靠得多.

2 算法设计

文献 [18] 给出了 TCP 流量控制的非线性模型, 而文献 [19] 用小信号理论在稳态工作点对其进行了局部线性化. 如果用 PID 控制器来设计作用在路由器上的队列管理算法, 那么整个 TCP 流量控制系统可以用图 1 所示的结构框图来描述. 建模和线性化的细节详见文献 [18] 和文献 [19], 图 1 中各参数的物理意义如下: C 为链路容量, N 为激活的 TCP 会话数, p 为分组丢弃概率, 即系统的控制量, q_0 为期望队列长度, R 为往返时延, $R = q/C + T_p$, 其中 T_p 为固定的广播时延. K_p , K_i 和 K_d 分别是 PID 控制器的比例、积分和微分常数.

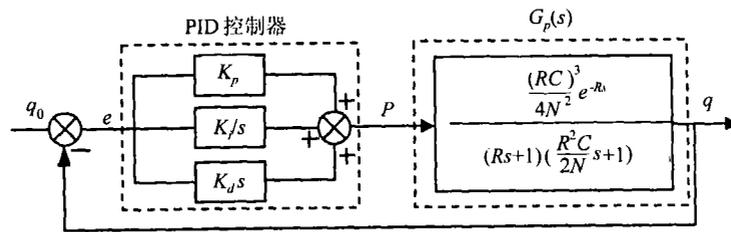


图 1 TCP 流量控制结构框图

路由器队列是一典型的离散系统, 我们需要对模拟 PID 控制算法进行必要的离散化处理, 以一系列的采样时刻点 kT 代表连续时间 t , 以和式代替积分, 以增量代替微分, 得到离散的 PID 表达式为

$$p(k) = K_p e(k) + K_i \sum_{j=0}^k e(j) + K_d \left\{ \frac{[e(k) - e(k-1)]}{T} \right\} \quad (1)$$

表示成增量形式:

$$\Delta p(k) = K_p \left\{ \left(1 + \frac{T}{T_i} + \frac{T_d}{T} \right) e(k) - \left(1 + \frac{2T_d}{T} \right) e(k-1) + \frac{T_d}{T} e(k-2) \right\} \quad (2)$$

其中 $T_i = K_p/K_i$, $T_d = K_p/K_d$, T 为采样时间.

为了方便比较, 这里我们沿用文献 [20] 中给定的采样频率, 即 160Hz。相应地, 采样时间 T 则为 0.00625s。至此, 剩余的工作既是如何整定参数 K_p , K_i 和 K_d 。如前所述, 因为很难确切地得到系统的临界放大倍数和临界振荡周期, 自然也就无法通过 Ziegler-Nichols 方法来整定控制器参数; 试凑法 [20] 缺乏可靠的理论依据, 无法保证系统的性能指标。下面我们给出一种 PID 控制器的参数整定方法, 使得闭环系统有确定的幅值和相角稳定裕度。

3 参数整定方法

如图 1 所示, 假定 PID 控制器作用下系统的幅值裕度和相角裕度分别为 A_m 和 P_m , 基于稳定性的要求, 应该有 $A_m > 1$ 或 $0 < P_m < \pi/2$ 。依据稳定裕度的定义, 我们有

$$G_p(j\omega_p)[K_p + j(K_d\omega_p - K_i/\omega_p)] = -1/A_m \quad (3)$$

$$G_p(j\omega_g)[K_p + j(K_d\omega_g - K_i/\omega_g)] = -e^{jP_m} \quad (4)$$

这里 ω_p 和 ω_g 分别为相角和幅值的截止频率。参数整定的目标就是通过调节 K_p , K_i 和 K_d 使 (3) 式和 (4) 式确定的 A_m 和 P_m 满足给定要求。

值得注意的是 (3) 式和 (4) 式中共有 5 个未知参数, 虽然可以将一个复数方程拆分为两个实数方程, 但未知数的个数依旧比方程数多 1, 因此需要增加一个额外的约束来保证唯一确定的解。在控制器的设计中, 闭环系统的带宽是应该仔细选择的, 太宽则控制信号容易饱和; 太窄则系统的响应缓慢。工程实践中, 一般选择闭环系统的带宽接近于开环带宽。因为控制器主要作用在相角截止频率以下的范围内, 通常用它近似闭环系统的带宽。所以, 我们可以作如下选择:

$$\omega_p = \alpha\omega_c, \quad \alpha \in [0.5, 2] \quad (5)$$

α 的缺省值为 1, ω_c 为被控对象的相角截止频率, 它满足

$$\angle G_p(j\omega_c) = -\pi \quad (6)$$

联立方程组 (3), (4), (5) 和 (6) 式, 解得

$$K_p = \operatorname{Re}[-1/(A_m G_p(j\omega_p))] \quad (7)$$

$$K_i = (x_p\omega_g - X_g\omega_p)(\omega_p/\omega_g - \omega_g/\omega_p)^{-1} \quad (8)$$

$$K_d = (X_p/\omega_g - X_g/\omega_p)(\omega_p/\omega_g - \omega_g/\omega_p)^{-1} \quad (9)$$

这里, $X_p = \operatorname{Im}[-1/(A_m G_p(j\omega_p))]$, $X_g = \operatorname{Im}[-e^{jP_m}/G_p(j\omega_g)]$ 。 ω_g 满足 $\operatorname{Re}[-e^{jP_m}/G_p(j\omega_g)] = \operatorname{Re}[-1/(A_m G_p(j\omega_p))]$, 即

$$\frac{|G_p(j\omega_p)| \cos(P_m - \angle G_p(j\omega_p))}{|G_p(j\omega_g)| \cos(\angle G_p(j\omega_p))} = \frac{1}{A_m} \quad (10)$$

假定图 1 中, 链路容量 C , 往返时间 R 和连接数 N 分别为 10^5 packet/s, 0.03s 和 30, 那么:

$$G_p(s) = 7.5 \times 10^6 e^{0.03s} / [(0.03s + 1)(1.5s + 1)] \quad (11)$$

依据 (7), (8), (9) 和 (10) 式, 选择 $\alpha = 1$, 给定闭环系统的幅值裕度和相角裕度分别为 3 和 $\pi/3$, 我们可以得到 $K_p = 2.583911 \times 10^{-6}$, $K_i = 2.982912 \times 10^{-5}$, $K_d = 3.507352 \times 10^{-8}$ 。

设定采样时间 T 为 0.00625s, 由 (2) 式得到用于队列管理的 PID 控制器的离散增量表达式为

$$p(k+1) = p(k) + 6.38956 \times 10^{-6} e(k) - 9.59862 \times 10^{-6} e(k-1) + 3.50735 \times 10^{-6} e(k-2)$$

4 仿真研究

下面我们用仿真试验来评估我们得到的 PID 控制器的性能, 仿真平台采用 NS-2^[21], 仿真网络的拓扑结构如图 2 所示. 瓶颈链路位于节点 A 和节点 B 之间, 链路容量 15Mbps (3750packet/s, 分组缺省大小为 500byte), 延时 15ms; 所有业务源均为持久性的 FTP 业务源, 它们与节点 A 之间的链路容量均为 10Mbps, 延时 15ms; 除节点 A 的队列由 PID 或 PI^[22] 控制器控制外, 其余队列均为 DropTail, 所有节点的缓存大小均为 300packet. 给定连接数 $N = 120$, 路由器队列长度的期望值为 100packet, 得到如图 3 和图 4 所示的仿真结果, 不难看出 PID 控制器迅速地将队列长度调节到了期望值附近, 但 PI 控制器却需要花费相当长的时间, 这一点图 4 中分组丢弃概率曲线的变化过程反映的更加明显. 由图 4 可以近似得到 PID 和 PI 控制器的调节时间分别为 5s 和 20s. 队列管理算法响应的迟缓性对实现主动队列管理的技术目标是不利的, 主动队列管理主要关心如何在保持高链路利用率的前提下, 控制分组在路由器队列中的排队时间. 如果链路相应的出口缓存队列一直保持非空, 便可以达到 100% 的链路利用率. 因此, 在任何负载状态下, 将队列长度保持在一个期望值附近, 上述的两个技术目标也就自然而然地实现了. 队列长度的波动必然导致排队时延的抖动, 这在强调 QoS 的网络技术研究背景下是应该避免的. 为了进一步说明瞬态响应迟缓性在负载瞬息万变的网络环境下对控制排队时间带来的负面影响, 我们进行下面仿真试验. 将 270 个业务源分为三组, 0 时刻启动第一组; 20s 之后启动第二组; 再隔 20s 启动第三组. 仿真结果记录在图 5 和图 6 中. 很显然, PID 控制的队列对负载变化的敏感程度没有 PI 那么强, 很快便将队列调节到了期望值附近, PI 则不然, 图 6 中关于控制量 (分组丢弃概率) 的调节过程也证实了这一点.

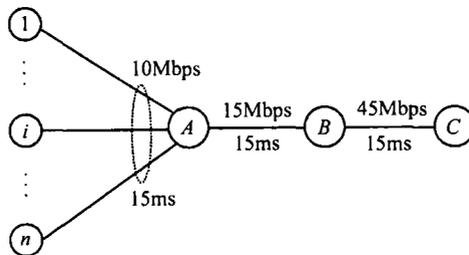


图 2 仿真网络拓扑结构

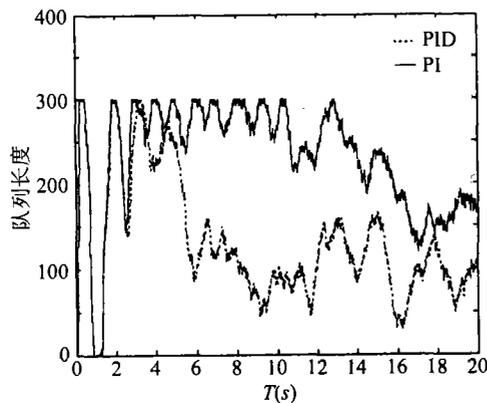


图 3 恒定负载队列 ($N = 120$)

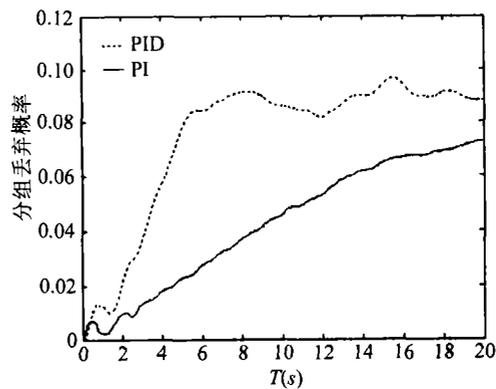
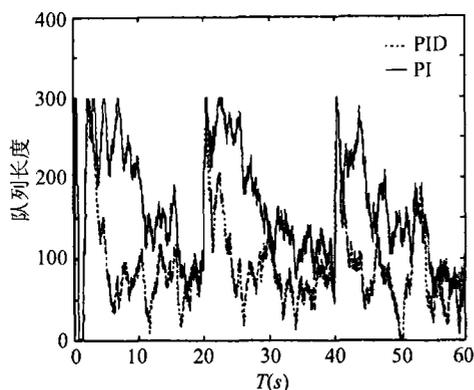
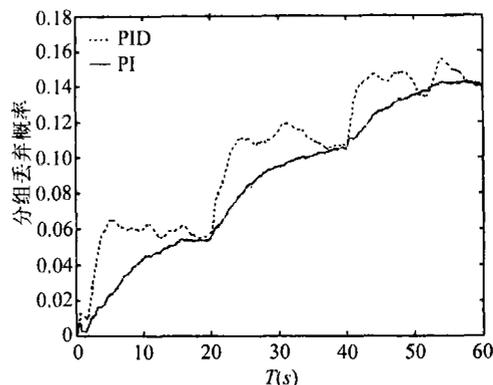


图 4 分组丢弃概率 ($N = 120$)

图5 可变负载队列 ($N = 270$)图6 分组丢弃概率 ($N = 270$)

5 结 论

主动队列管理是作用在网络中间节点上的一种非常有效的拥塞控制策略,在保证较高吞吐量的同时,有效地控制了缓存队列的长度,从而为要求时延保证的业务提供一种良好的实现机制。C. Hollot 等人提出的 PI 控制算法可以将队列长度控制在任意的期望值附近,从而使终端系统很容易控制分组在网络中的排队时延。在参数的整定上,PI 控制器采用了试凑法,瞬态和稳态性能得不到可靠保障。在本研究中,我们为了克服 PI 控制算法响应迟缓的弱点,引入微分环节来增强系统的响应能力,并给出了基于稳定裕度的参数整定方法,使得到的 PID 控制器的稳定性有了可靠保障。仿真试验表明 PID 控制器的瞬态性能指标较 PI 算法有显著的提高,为在负载瞬息万变的网络环境中实现控制分组排队等待时间的目标提供了有力的技术保障。

参 考 文 献

- [1] V. Jacobson, M. J. Karels, Congestion avoidance and control, In Proc. of the ACM SIGCOMM'88 Conference, Stanford, CA, USA, 1988, 314-329.
- [2] W. Stevens, TCP slow start, congestion avoidance, fast retransmit, and fast recovery. RFC 2001, <http://www.rfc.net/rfc2001.html>.
- [3] S. Floyd, The addition of Explicit Congestion Notification(ECN) to IP, <http://www.aciri.org/floyd/papers.html>.
- [4] A. ManKim, K. Ramakrishnan, Gateway congestion control survey, RFC1254, <http://www.rfc.net/rfc1254.html>.
- [5] B. Braden, *et al.*, Recommendations on queue management and congestion avoidance in the Internet, RFC2309, <http://www.rfc.net/rfc2309.html>.
- [6] S. Floyd, V. Jacobson, Random early detection gateways for congestion avoidance, IEEE/ACM Trans. on Networking, 1993. 1(4), 397-413.
- [7] S. Floyd, A report on some recent development in TCP congestion control, <http://www.aciri.org/floyd/papers.html>.
- [8] M. Christiansen, K. Jeffay, D. Ott, F. D. Smith, Tuning RED for Web traffic, In Proc. of the ACM SIGCOMM 2000 Conference, Stockholm, Sweden, 2000, 139-150.
- [9] V. Firoiu, M. Borden, A study of active queue management for congestion control, In Proc. of INFOCOM2000 Conference, Tel Aviv, Israel, 2000, 1435-1444.
- [10] M. May, T. Bonald, T. Bolot, Analytic evaluation of RED performance. In Proc. of INFOCOM2000 Conference, Tel Aviv, Israel, 2000, 1415-1424.
- [11] S. Folyd, K. Fall, Promoting the use of end-to-end congestion control in the Internet, IEEE/ACM Trans. on Networking, 1999, 7(4), 458-472.

- [12] Cisco, Distributed weighted random early detection, <http://cco.cisco.com>.
- [13] W. Feng, D. Kandlur, D. Saha, K. Shin, Blue: A new class of active queue management algorithm, University of Michigan Technical Reports CSE-TR-387-99, April, 1999.
- [14] T. J. Ott, T. V. Lakshman, L. H. Wong, SRED: Stabilized RED, In Proc. of INFOCOM1999 Conference, New York, NY, USA, 1999, 1346-1355.
- [15] D. Lin, R. Morris, Dynamics of random early detection, In Proc. of the ACM SIGCOMM 1997 Conference, Cannes, France, 1997, 139-150.
- [16] W. Feng, D. Kandlur, D. Saha, K. Shin, A self-configuring RED gateway, Proc. INFOCOM'99, New York, NY, USA, 1999, 1320-1328.
- [17] F. Anjum, L. Tassiulas, Balanced-RED: An algorithm to achieve fairness in Internet, <http://www.isr.umd.edu/CSHCN/>.
- [18] C. Hollot, V. Misra, D. Towsley, W. B. Gong, A control theoretic analysis of RED, In Proc. of INFOCOM2000 Conference, Tel Aviv, Israel, 2000.
- [19] V. Misra, W. B. Gong, D. Towsley, Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED, In Proc. of the ACM SIGCOMM 2000 Conference, Stockholm, Sweden, 151-160.
- [20] C. Hollot, V. Misra, D. Towsley, W. B. Gong, On designing improved controllers for AQM routers supporting TCP flows, In Proc. of INFOCOM2000 Conference, Tel Aviv, Israel, 2000.
- [21] UCN/LBL/VINT, Network Simulator-NS2, <http://www-mash.cs.berkeley.edu/ns>.

PID CONTROLLER FOR ACTIVE QUEUE MANAGEMENT

Ren Fengyuan Wang Fubao* Ren Yong Shan Xiuming

(Department of Electronic Engineering, Tsinghua University, Beijing 100084, China)

*(Dept. of Computer Sci. and Eng., Northwestern Polytechnical Univ., Xi'an 710072, China)

Abstract Acting on the intermediate nodes, Active Queue Management (AQM) either keeps the high throughput for routers or effectively controls the queue lengths, so that arrives the object to control end-to-end delay for QoS guarantees. C. Hollot, *et al.* (2001) designed the PI controller for AQM with the frequency approach in classical control theory, but it seems to be unscientific to tune the controller parameters through trial-errors, moreover the transient performance of the PI controller is not perfect, such as the adjusting time is very long, etc. In this paper, the differential component is introduced to enhance the responsibility, and an approach for tuning controller parameters is put forward based on the explicit gain and phase margins. The simulation results show that the integrated performance of PID controller designed with the new approach is obviously superior to that of the PI controller.

Key words Active queue management, PID controller, Congestion control

任丰原: 男, 1970年生, 博士, 主要研究领域为网络流量管理及其优化, 测控网络等。
王福豹: 男, 1964年生, 博士, 教授, 主要研究方向为计算机网络和媒体流技术。
任勇: 男, 1962年生, 博士, 教授, 主要研究领域为人工生命, 网络流量工程等。
山秀明: 男, 1946年生, 博士生导师, 教授, 主要研究领域为复杂系统理论及其在通信网中的应用。