

文章编号: 1001-0920(2006)08-0937-04

大时滞网络自适应预测 PI 主动队列管理算法

钱艳平, 李 奇

(东南大学 自动化研究所, 南京 210096)

摘 要: 针对网络中存在的大时滞和网络参数时变问题, 提出一种自适应预测 PI 主动队列管理算法。将 Smith 预估器与达林算法相结合, 既克服了大时滞带来的不利影响, 也减少了控制器参数整定数量。利用网络参数与控制参数所具有的确切关系, 通过在线估计网络参数来实时调节控制参数, 使得控制器能够适应网络参数的变化, 同时采用线性化方法分析了系统局部稳定性。仿真结果表明, 所提出的算法是可行而有效的。

关键词: 主动队列管理; 拥塞控制; 大时滞网络; 自适应预测 PI

中图分类号: TP393

文献标识码: A

Adaptive Predictive PI Algorithm for Active Queue Management in Large Delay Network

QIAN Yan-ping, LI Qi

(Research Institute of Automation, Southeast University, Nanjing 210096, China Correspondent: QIAN Yan-ping, E-mail: qyp-wxf@163.com)

Abstract An adaptive predictive proportional integral (APPD) algorithm for active queue management (AQM) is proposed to cope with the large delays and time-varying network parameters. Smith predictor and Dahlin algorithm are combined to alleviate the unfavorable influences of time-delay and reduce the number of the tuning parameters. The control parameters are automatically tuned according to on-line estimation of link capacity and traffic load, which makes APPD perform well for a wide range of network conditions. The local stability of the system is analyzed using the linearization method. Simulation results show the effectiveness of the APPD algorithm.

Key words: Active queue management; Congestion control; Large delay network; Adaptive predictive PI

1 引 言

随着计算机网络规模的不断扩大, 拥塞控制在保证网络正常运行方面变得日益重要。在网络端实行主动队列管理(AQM)拥塞避免策略已成为一个活跃的研究领域^[1]。目前, 主要 AQM 算法包括 RED (Random early detection) 算法^[2]和比例积分(PD)算法^[3]等。

在网络端采用 AQM 算法能较大地改进拥塞控制的性能。但是, 目前的 AQM 策略存在两个主要问题: 首先是网络参数时变问题, 该问题为控制算法参数设计带来了很大的困难, 如 ARED (Adaptive RED)^[4]和 STAQM (Self-tuning AQM)^[5]等; 其次

是网络存在较大时延^[6], 如果采用常规控制器, 则可能导致系统不稳定。文献[7]利用 Smith 预估器控制网络流量, 取得了较好效果, 但算法调节参数较多且存在耦合。

本文提出一种自适应预测 PI (APPD) 控制算法。该算法首先将 Smith 预估器与达林算法相结合, 设计了一种预测 PI 控制器 (PPD)^[8], 既减少了需要整定的参数数量, 也避免了参数整定时相互影响。同时, 通过对网络参数的在线估计, 实时调整控制参数, 使得控制器能够适应网络参数的变化。仿真结果表明, 本文算法是可行而有效的。

2 TCP/AQM 控制理论模型

AQM 作用下长持续 (Long lived) TCP 流控制

收稿日期: 2005-05-26; 修回日期: 2005-08-18

作者简介: 钱艳平(1975—), 男, 浙江舟山人, 博士生, 从事网络拥塞控制、智能控制等研究; 李奇(1963—), 男, 江苏苏州人, 教授, 博士生导师, 从事网络拥塞控制、智能控制、最优控制等研究。

理论模型如图1所示,建模及线性化过程详见文献[3,9]图1中: q_s 为期望队列长度, q 为拥塞队列长度, R 为往返传播时延(RTT), C 为链路容量, M 为负载因子(TCP会话数), p 为分组丢弃概率

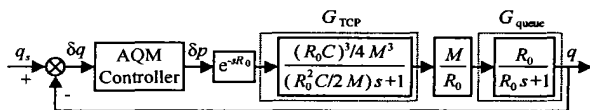


图1 TCP/AQM 线性化控制系统结构图

由图1可得 TCP/AQM 系统的时滞二阶模型

$$G(s) = \frac{K e^{-\tau s}}{(T_1 s + 1)(T_2 s + 1)} \quad (1)$$

其中

$$K = (R_0 C)^3 / (2M)^2, T_1 = R_0,$$

$$T_2 = R_0^2 C / (2M), \tau = R_0$$

式(1)中一般有 $T_2 \gg T_1$ 。例如网络参数为 $R_0 = 0.467$ s, $C = 15$ Mbps, $M = 120$, 分组长度为 500 bytes 时, 可得 $T_1 = 0.467$, $T_2 = 3.40$ 。因此, 为使控制器设计简单方便, 将式(1)进一步简化为一阶模型

$$G(s) \approx G_0(s) e^{-\tau s} \quad (2)$$

其中: $G_0(s) = K / (T s + 1)$, $T = T_1 + T_2$

3 自适应预测 PI 控制器设计

针对网络中的大时滞和参数时变, 首先用达林算法设计 Smith 预估器中的控制器; 然后结合网络参数估计设计 APPI 算法。系统结构如图2所示, 其中: $G_m(s) e^{-(\tau + \xi)s}$ 为预估对象模型, ξ 为时滞误差

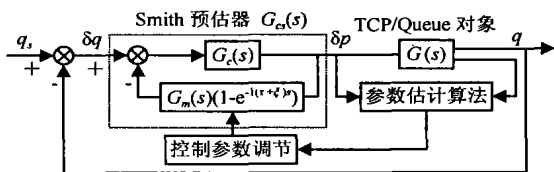


图2 基于自适应预测 PI 算法的系统框图

3.1 预测 PI 控制算法

网络拥塞控制具有可以通过机理分析, 在网络参数确定时获得比较准确的系统对象模型的优点, 因此选用 Smith 预估器来解决系统中的时滞

采用达林算法设计 Smith 预估器中的控制器 $G_c(s)$ 。假定期望闭环传递函数为 $W(s) = e^{-\tau s} / (T_g s + 1)$, 其中 T_g 为期望闭环响应时间常数, 有如下定理^[8]:

定理1 在满足 $G_m(s) = G_0(s)$ 和 $\xi = 0$ 模型匹配条件下, 针对式(2)所示时滞对象, 采用达林算法设计 Smith 预估器中的控制器 $G_c(s)$ 为 PI 控制器, 即 $G_c(s) = K_p [1 + 1 / (T_i s)]$, 且控制参数为 $K_p = T / (K T_g)$, $T_i = T$ 。

由定理1可知, Smith 预估器输出为

$$u(t) = K_p \delta q + \frac{K_p}{T_i} \int_0^t \delta q dt - \frac{1}{T_g} \int_0^t [u(t) - u(t - \tau)] dt, \quad (3)$$

则网络链路队列分组丢弃概率 $p(t)$ 为

$$p(t) = \min\{1, \max\{0, u(t)\}\}. \quad (4)$$

3.2 控制参数自适应算法

由定理1可知, 控制器参数与网络参数有确切关系, 因此可通过估计网络参数 (N, RTT, C) 来估计对象参数, 进而修正控制器控制参数。假设网络参数变化的时间粒度 (Time scale) 远大于控制器响应时间粒度, 即 PPI 算法控制系统达到平衡点时, 网络参数近似为恒值。同时, 对于大时滞网络, 排队时延相对于传播时延可以忽略, 因此可用传播时延替代网络参数中的 RTT, 这里假定传播时延已知

本文采用文献[5]的方法估计链路可用带宽和网络负载

3.2.1 链路可用带宽估计

链路可用带宽可通过定期计算路由器发送的 TCP 分组数获得, 即

$$C = N_{tcp} / T_{sam} \quad (5)$$

其中: \hat{C} 为链路可用带宽, T_{sam} 为链路带宽计算周期; N_{tcp} 为链路带宽计算周期内发送的 TCP 分组数

采用低通滤波器处理 TCP 可用带宽因受其他业务量的干扰而发生的波动为

$$\dot{\hat{\theta}} = -K_c \hat{\theta} + K_c \hat{C} \quad (6)$$

其中: $\hat{\theta}$ 为估计的链路带宽, K_c 为滤波器时间常数

3.2.2 TCP 负载估计

当系统处于平衡点时, 由网络流量模型可知^[3]

$$M / R_0 C = \sqrt{p_0 / 2} \quad (7)$$

式(7)是 TCP 的固有性质, 与所采用的 AQM 策略无关。虽然式(7)仅在平衡点成立, 但分组丢弃概率可在路由器中方便获得, 因此可用来近似估计暂态时的 $M / (R_0 C)$, 并采用低通滤波器平滑处理

$$\dot{\hat{\theta}} = -K_l \hat{\theta} + K_l \sqrt{\bar{p}} / 2 \quad (8)$$

其中: $\hat{\theta}$ 为 $M / (R_0 C)$ 的暂态估计值; K_l 为滤波器时间常数; \bar{p} 为自适应更新周期内平均分组丢弃概率, 在稳定性分析时用 \bar{p} 代替 p 。

3.2.3 控制参数自适应

PPI 控制器的控制参数与网络参数有确定的关系。由定理1和式(1), (2)可知, 控制器比例系数、积分系数与网络参数的关系为

$$\begin{cases} K_p = \frac{R_0 + R_0^2 C / (2M)}{(R_0 C)^3 / (2M)^2 \cdot T_g} \\ K_i = K_p / T_i = 1 / [(R_0 C)^3 / (2M)^2 \cdot T_g] \end{cases} \quad (9)$$

利用链路带宽、网络负载估计值, 可以在线修正 PPI 控制器控制参数

$$\begin{cases} K_p^0 = (4\theta + 2\theta) / (\theta \cdot T_g), \\ K_i^0 = 4\theta / (R_0 T_g \theta). \end{cases} \quad (10)$$

4 稳定性分析

4.1 网络参数固定

网络参数固定时自适应算法不起作用, 系统稳定性由对象和控制器决定。在预估对象模型与实际对象模型(2)完全匹配时, 可得系统特征方程为 $1 + T_g s = 0$, 所以对任意 $T_g > 0$, 控制系统均是渐近稳定的。当 T_g 较小时, 闭环响应较快, 但系统鲁棒性较差; 当 T_g 较大时, 闭环响应较慢, 但鲁棒性较好。

4.2 网络参数变化

采用在平衡点线性化方法分析系统的局部稳定性。为分析方便, 将 PPI 控制器改写为

$$p(t) = K_p \dot{\alpha}_q + K_p / T_i \cdot x(t) - 1/T_g \cdot y(t). \quad (11)$$

其中: $\dot{x}(t) = \dot{\alpha}_q, \dot{y}(t) = p(t) - p(t - \tau)$ 。

式(6), (8) 及(11) 的平衡点满足

$$\begin{aligned} \theta &= \hat{C}, \theta = \sqrt{p_0/2}, q = q_s, \\ p_0(t) &= p_0(t - \tau), \\ p_0(t) &= K_{i0} x_0 - 1/T_g \cdot y_0 \end{aligned} \quad (12)$$

$p(0) = 0$, 且分组丢弃概率一般 $p(t) \ll 1$, 因此, 为分析方便, 假设 $y_0 = 0$ 则在平衡点线性化后得

$$\begin{cases} \dot{\alpha} = -K_c \alpha, \\ \dot{\alpha} = -K_i \alpha + K_i / (2\sqrt{2}\sqrt{p_0}) \cdot \dot{\phi}, \\ \dot{\alpha} = \dot{\alpha}_q, \dot{\delta}_y = \dot{\phi}(t) - \dot{\phi}(t - \tau), \\ \dot{\phi} = \left(\frac{\partial K_i^0}{\partial x} \cdot x \right) \Big|_0 \alpha + \left(\frac{\partial K_i^0}{\partial x} \cdot x \right) \Big|_0 \alpha + \\ K_{p0}^0 \dot{\alpha}_q + K_{i0}^0 \dot{\alpha} - 1/T_g \cdot \dot{\delta}_y. \end{cases} \quad (13)$$

经计算, 可以得到

$$\dot{\phi} = 2\sqrt{2p_0} \alpha - p_0/C \cdot \alpha + K_{p0}^0 \dot{\alpha}_q + K_{i0}^0 \dot{\alpha} - 1/T_g \cdot \dot{\delta}_y. \quad (14)$$

将上式代入式(13), 可得

$$\begin{aligned} \dot{\alpha} &= K_i / (2\sqrt{2p_0}) \left[-\frac{p_0}{C} \alpha + K_{p0}^0 \dot{\alpha}_q + \right. \\ &\quad \left. K_{i0}^0 \dot{\alpha} - \frac{1}{T_g} \dot{\delta}_y \right] \end{aligned} \quad (15)$$

将式(13) ~ (15) Laplace 变换, 与式(2) TCP 以及队列模型结合, 可得图 3 所示的系统结构

整个系统的特征方程为

$$T_g s^2 + s + K_i = 0, \quad (16)$$

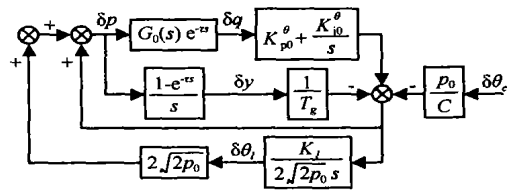


图 3 采用 APPI 算法的线性化控制系统结构
则系统为局部渐近稳定只要系统满足

$$K_i > 0, T_g > 0 \quad (17)$$

对于 K_i 的选择, 考虑到设计控制参数自适应算法时的基本假设(假设网络参数变化的时间粒度远大于控制器响应时间粒度), 可通过适当选取较小的 K_i 值来满足。根据经验, 可选择满足 $0 < K_i < 0.25\tau$, 且 $K_i < 1$ 的合适值。

5 算法仿真

采用网络仿真软件 ns-2^[10] 验证算法有效性。网络拓扑如图 4 所示, n_1 与 n_2 之间为瓶颈链路, n_2 与 n_3 之间传播时延为 $d = 180 \text{ ms}$, 分组长度为 500 bytes, 队列容量为 800 packets; 除瓶颈链路外, 其他链路均采用 Drop Tail 算法。设期望队列长度为 250 packets。网络参数为 $C = 15 \text{ Mbps}$, $M = 120$, $RTT = 0.467 \text{ s}$, 采样频率为 160 Hz。PI 采用默认参数, APPI 控制参数 $\tau = 0.42$, $K_c = 0.42$, $K_i = 0.1$, 自适应更新周期为 0.3 s。

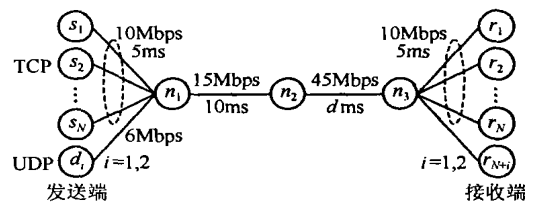


图 4 网络拓扑结构

实验 1 验证网络参数固定时 ARED, PI 和 APPI 算法控制瓶颈链路队列效果, 仿真结果见文献 [8]。当大时延存在时, 使用 RED 算法(包括 ARED), 队列振荡极大, 已不适用于 AQM 控制; 使用 PI, 队列长度变化较大; 采用 APPI, 队列长度变化较小, 基本围绕给定 250 packets 变化。

实验 2 验证网络参数变化时 PI 和 APPI 算法的控制效果。初始网络参数、控制参数同实验 1。

图 5~ 图 7 分别描述了 M, C 变化和存在不可控流干扰时, 瓶颈链路使用 PI 和 APPI 算法队列长度变化曲线。由图 5 可以看出, $t = 100 \text{ s}$ 时 $M = 720$, $t = 200 \text{ s}$ 时 $M = 50$, M 变化时 PI 算法收敛时间约需 40 s, 而 APPI 只约需 15 s, 且 M 增加使 APPI 队列更加稳定、振幅变小。由图 6 可以看出, $t = 150 \text{ s}$ 时 $C = 90 \text{ Mbps}$, C 增加后, 使用 PI 算法队列几乎为空, 导致链路利用率下降, 而 APPI 算法仍能保持队列长

度控制在 250 packets, 尽管振荡较大 图 7 用 UDP/CBR 流来模拟不可控流, 分别在 $t = 60$ s 和 $t = 120$ s 时加入发送速率为 6M bps 的 d_1, d_2 流, 在 $t = 180$ s 和 $t = 240$ s 时, 依次停止 d_1, d_2 流的发送 可以发现, 使用 APPI 算法的队列对于干扰引起的瓶颈链路可用带宽变化能够及时跟踪, 而使用 PI 算法的队列控制效果较差

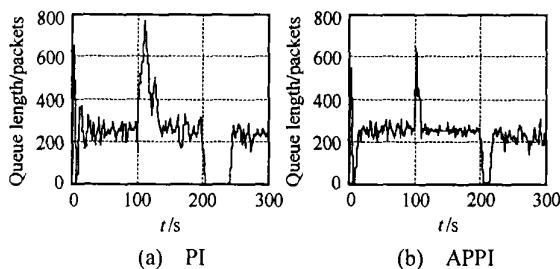


图 5 M 变化时瓶颈队列长度曲线

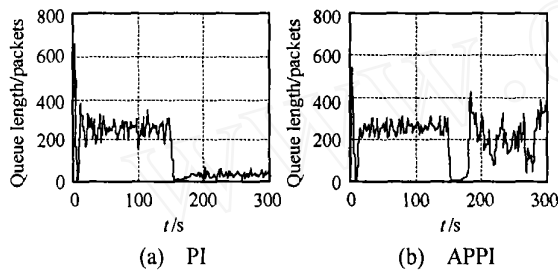


图 6 C 变化时瓶颈队列长度曲线

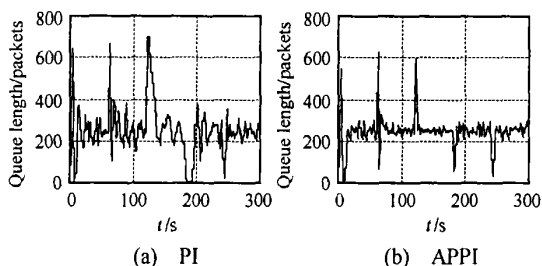


图 7 存在不可控流时瓶颈队列长度曲线

由实验 2 可知, 网络参数发生剧烈变化时(变化范围为初始值的 6 倍), APPI 仍能使系统保持稳定, 并获得了满意的控制效果

6 结 论

目前许多 AQM 策略未考虑到网络时延和网络参数的变化, 降低了系统稳定性和控制品质 本文提

出了一种自适应预测 PI 控制算法 理论分析和仿真结果表明, APPI 算法能够适应网络环境的动态变化和网络时延, 具有比 PIARED 等算法更好的控制效果, 但该算法的全局稳定性分析尚有待进一步的研究

参考文献 (References)

- [1] Braden B, Clark D, Crowcroft J, et al. *Recommendations on Queue Management and Congestion Avoidance in the Internet* [DB/OL]. <http://www.faqs.org/rfcs/rfc2309>, 1998
- [2] Floyd S, Jacobson V. Random Early Detection Gateways for Congestion Avoidance [J]. *IEEE/ACM Trans on Networking*, 1993, 1(4): 397-413
- [3] Misra V, Towsley D, Gong W. Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED [A]. *Proc of ACM SIGCOMM 2000* [C]. Stockholm, 2000: 151-160
- [4] Floyd S, Gummadi R, Shenker S. *Adaptive RED: A New Algorithm for Increasing the Robustness of RED* [DB/OL]. <http://www.icir.org/floyd/papers/adaptivered.pdf>, 2001
- [5] Zhang H, Hollot C V, Towsley D, et al. A Self-tuning Structure for Adaptation in TCP/AQM Network [A]. *Proc of GLOBECOM '03* [C]. San Francisco, 2003: 3641-3646
- [6] Barakat C. TCP/IP Modeling and Validation [J]. *IEEE Network*, 2001, 15(3): 38-47
- [7] Mascolo S. Congestion Control in High-speed Communication Networks Using the Smith Principle [J]. *Automatica*, 1999, 35(12): 1921-1935
- [8] Qian Y P, Li Q, Diao X. A Predictive PI Algorithm for Active Queue Management of Delay Network [A]. *Proc of ICCAS '05* [C]. Hong Kong, 2005: 95-99
- [9] Hollot C V, 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
- [10] UCN / LBL / VNT. *Network Simulator - NS2* [DB/OL]. <http://www.isi.edu/nsnam/ns>, 1995