基于粒子群优化的网络拥塞控制新算法

陆锦军[1,2],王执铨[2]
(1. 南通职业大学现代教育技术中心,江苏南通 226007;2. 南京理工大学自动化学院,江苏南京 210094)

摘 要: PI 控制器常用于主动队列管理中,但参数整定上的试凑法具有盲目性,算法的瞬态性能也不够理想. 本文推导了基于流体流理论的网络简化模型,基于该模型将集群智能中的改进粒子群优化算法(PSO)应用于 PID 控制器参数优化,定义了一个综合调节时间、上升时间、超调量、系统静态误差、正弦跟踪误差等动静态性能指标函数,在给定的参数空间进行组合优化搜索,迅速求得获取使性能指标优化函数极小化的一组 PID 控制器参数,将 PID 控制器应用于网络主动队列管理系统中. 仿真结果表明,在大时滞和突发业务流的冲击两种情况下,该方法设计的控制器的动静态性能优于 RED、PI 算法,超调量均小于 5 %,调节时间分别小于 5 秒、4 秒,稳态误差分别小于两个数据包和 3 个数据包.

关键词: 主动队列管理; 网络拥塞; PID 控制; 粒子群优化
中图分类号: TP273 **文献标识码**: A **文章编号**: 0372-2112 (2007) 08-1446-06

A New Network Cogestion Control Algorithm Based on Particle Swarm Optimization

LU Jin-jun[1,2], WANG Zhi-quan[2]
(1. *Center of Education and Technology, Nantong Vocational College, Nantong, Jiangsu* 226007, *China*;
2. *School of Automation, Nanjing University of Science and Technology, Nanjing, Jiangsu* 210094, *China*)

Abstract: PI controller is often used to control active queue management (AQM), but its trial method of tuning controller is aimless, and the dynamic performance of algorithm is not enough satisfying. Simplified network model based on fluid flow theory is derived in this paper, and based on this model, an improved algorithm, i.e. particle swarm optimization (PSO) algorithm is applied to optimization of PID controller parameters. In the following, a new performance function including the system adjusting time, rise time, overshoot, steady state error and sinusoidal position tracking error is defined. It is fast to calculate a group of PID controller parameters that minimize the evaluation function by searching in the given controller parameter area, and then the PID controller is applied to AQM system. The simulation experimental results show that under the two conditions of large time delay and sudden business flow, the overshoot is both less than 5 %, the adjusting time is less than 5 seconds and 4 seconds separately, and the steady error is less than 2 packets and 3 packets separately, so the dynamic state and steady state performances of the proposed algorithm are obviously superior to those of the existing RED and PI algorithms under the two conditions.

Key words: active queue management; network congestion; PID control; particle swarm optimization

1 引言

IP 网络拥塞控制是人们一直着力解决但未能很好解决的问题,相继产生了不少有影响力的算法,如 RED[1]、ARED[2]、SRED[3]、BLUE[4]等,同时也出现了许多基于网络流量的控制模型,但较具影响力的是 V Misra 等人于 2000 年基于流体流理论提出的网络模型[5],该模型较为恰当地描述了 TCP 传输流的行为[6],为研究人员广为采用,根据该模型,产生了 PI[7]等主动队列管理算法,增强了对队列长度的控制能力,由于在动态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 方法来整定控制参数,被迫采用了试凑整定方法,理论分析和实验仿真证明 PI 控制器调节时间过长,对路由器缓存大小的依赖过强. 文献[8]提出了一种基于速度控制新的 API 网络拥塞控制策略,与 PI 控制器相比,这种算法提高了在较少目标队列长度下的收收敛速度,响应速度,但在流量突发时,队列长度具有一定的抖动.

收稿日期:2006-11-16;修回日期:2007-04-10

粒子群算法 PSO 是由 Kennedy 等人于 1995 年提出的一种新的全局优化进化算法,它源于对鸟类捕食行为的模拟[9],是基于群体智能理论的优化算法,通过群体中粒子间的合作与竞争产生的群体智能指导优化搜索,具有个体数目少、计算简单、鲁棒性好等优点,目前该算法已经成功地应用于函数参数优化、神经网络训练、模糊系统控制等领域[10~14]. 本文推导了基于流体流理论的网络简化模型,基于该模型将 PID 控制器应用于网络主动队列管理系统中,应用集群智能中的改进粒子群优化算法(PSO)对控制器参数进行组合优化. 定义了一个综合上升时间、调节时间、超调量、系统静态误差、正弦跟踪误差等动静态性能指标函数,在给定的空间迅速搜索得使性能指标优化函数极小化的一组 PID 控制器参数. 仿真结果表明,在大时滞和突发业务流的冲击两种情况下,该方法设计的控制器的动静态性能优于 RED、PI 算法.

2　TCP/ AQM 简化模型及其 AQM 控制

V Misra 等人在分析网络连续数据流和随机微分方程的基础上,建立了 TCP 的动态模型[6],用如下一组非线性微分方程来描述.

$$\dot{W}=\frac{1}{R(t)}-\frac{W(t)\,W(t-R(t))}{2\,R(t-R(t))}\,p(t-R(t)) \tag{1}$$

$$\dot{q}=\frac{N(t)}{R(t)}\,W(t)-C(t) \tag{2}$$

式中:W 为预期的 TCP 拥塞窗口的大小(包);q 为预期的队列长度(包);$R(t)$ 为往返时间;$R(t)=\frac{q(t)}{C(t)}+T_p$(秒),$T_p$ 为传输延时(秒);C 为链路容量(包/秒);N 为激活 TCP 连接数;P 为分组的丢弃概率,P 的取值范围为$[0,1]$;q 和 W 满足 $q\in[0,\bar{q}]$,$W\in[0,\bar{W}]$. 其中,$\bar{q}$、$\bar{W}$ 分别表示缓存容量和最大窗口尺寸. 式(1)中第一个方程描述的是 TCP 的窗口控制动态特性. 第二个方程描述的是瓶颈队列长度,它等于包到达率 NW/R 和链路容量 C 之间的差值. 用方框图 1 表示式(1)微分方程.

系统正常工作在稳定工作点附近,为了更好的分析反馈控制 AQM,利用稳定工作点附近的小信号线性化处理式(1).

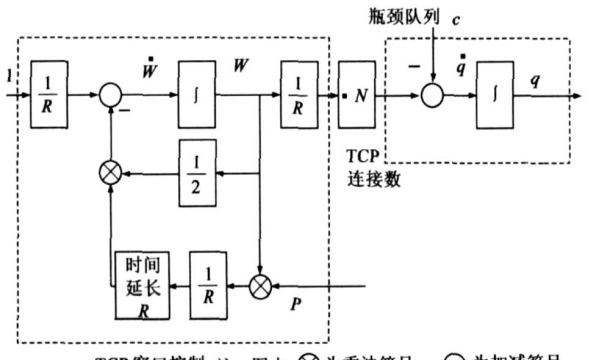

图 1　TCP 拥塞避免控制图

假设 TCP 会话数和链路容量是常量,即 $N(t)\equiv N$,$C(t)\equiv C$. 取(W,q)为状态,p 为输入,则通过定义 $\dot{W}=0$ 和 $\dot{q}=0$ 可求得工作点(W_0,q_0,p_0):

$$\dot{W}=0\Rightarrow W_0^2p_0=2 \tag{3}$$

$$\dot{q}=0\Rightarrow W_0=\frac{R_0C}{N};\ R_0=\frac{q_0}{C}+T_p \tag{4}$$

另外忽略时间延迟 $t-R$ 对队列长度 q 的相关性,并假设它的值固定为 $t-R_0$. 另一方面,保留动态参数中往返时间对队列长度的相关性,由此得到简化的动态方程.

$$\dot{W}=\frac{1}{\frac{q(t)}{C}}-\frac{W(t)}{2}\,\frac{W(t-R_0)}{\frac{q(t-R_0)}{C}+T_p}\,p(t-R_0) \tag{5}$$

$$\dot{q}=\frac{N}{R(t)}\,W(t)-C \tag{6}$$

在工作点对式(2)进行线性化处理得:

$$\delta\dot{W}(t)=-\frac{N}{R_0^2C}(\delta W(t)+\delta W(t-R_0))-\frac{1}{R_0^2C}(\delta q(t)-\delta q(t-R_0))-\frac{R_0C}{2N^2}\delta p(t-R_0) \tag{7}$$

$$\delta\dot{q}(t)=\frac{N}{R_0}\delta W(t)-\frac{1}{R_0}\delta q(t) \tag{8}$$

其中 $\delta W=W-W_0$;$\delta q=q-q_0$;$\delta p=p-p_0$. 表示工作点附近的扰动值变量. 线性化动态特性如图 2 所示,为了分析设计 AQM 机制,明确定义了 TCP 窗口动态和队列动态控制.

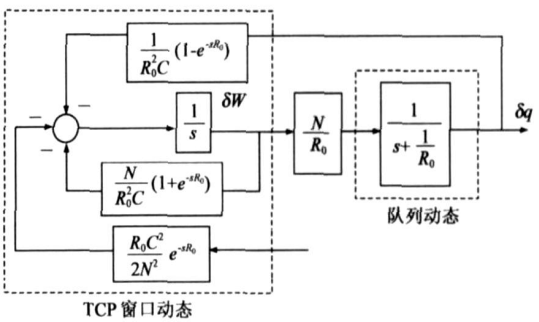

图 2　线性化 TCP 连接的方框图

为了分析方便,将窗口动态低频性能和高频性能分解得图 3. 其中,高频动态部分 $\Delta(s)$ 约等于 $\frac{2N^2s}{R_0^2C^3}(1-e^{-sR_0})$. 分析稳态工作点各参数之间的关系,主要研究低频性能,当拥塞控制达到稳态时,拥塞窗口应该远远大于 1,即 $W\gg1$ 时,$e^{-R_0s}\approx1$(除非业务流很多或网络负载能力较差,在稳态时对每个业务流分配到的带宽较小,拥塞窗口可能不会远远大于 1),忽略高频性能得到如图 4 所示的 TCP 连接的线性化简化框图. 在图 4 中

加入 AQM 控制,最终可得到如图 5 所示的基于简化模型 AQM 控制系统框图.

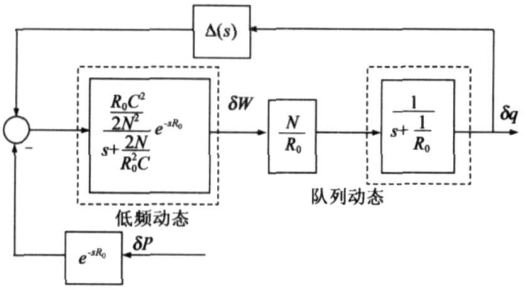

图 3 分离开低频窗口动态和高频部分的线性化动态方框图

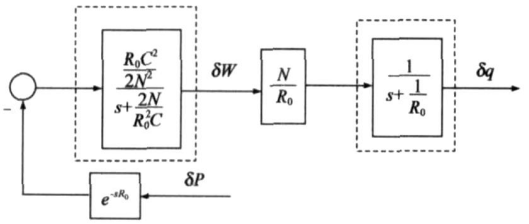

图 4 线性化 TCP 连接的简化框图

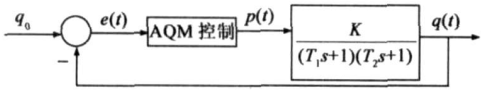

图 5 基于简化模型的 AQM 控制系统框图

其中 $k=\frac{(R_0C)^3}{4N^2}$, $T_1=R_0$, $T_2=\frac{R_0^2C}{2N}$.

令 $Gp(s)$ 为 AQM 系统简化模型,

即
$$Gp(s)=\frac{K}{(T_1s+1)(T_2s+1)} \tag{9}$$

若链路容量 C、往返时间 R_0 和连接数 N 分别为 10^5packet/s、0.03s 和 30,

则
$$Gp(s)=\frac{7.5\times10^6}{(0.03s+1)(1.5s+1)} \tag{10}$$

PID 控制是一种具有负反馈的闭环控制系统,能够较好的根据系统实时状态快速作出控制反应,故不妨假设图 5 中的 AQM 控制器仍具有 PID 形式,它引入微分环节来增强系统的快速响应的能力,克服其他控制算法响应迟缓的弱点,根据偏差的变化趋势调节,具有超前作用,对系统的时滞具有补偿能力.

即
$$G_c(S)=K_p+\frac{K_i}{s}+K_ds \tag{11}$$

其中 K_p、K_i、K_p 分别为 PID 控制器的比例、积分、微分增益系数,其离散的表达形式为

$$p(k)=K_pe(k)+K_1\sum_{j=0}^{k}e(j)+K_d\frac{[e(k)-e(k-1)]}{T} \tag{12}$$

其中 $e(k)=q(k)-q_0$, $q(k)$ 是第 k 时刻的队列长度采样值, q_0 为期望队列长度, $p(k)$ 为 k 时刻的丢包概率.

其增量形式为

$$\Delta p(k)=K_p\left\{\left(1+\frac{T}{T_i}+\frac{T_d}{T}\right)e(k)-\left(1+\frac{2T_d}{T}\right)e(k-1)+\frac{T_d}{T}e(k-2)\right\} \tag{13}$$

其中 $T_i=\frac{K_p}{K_i}$, $T_d=\frac{K_d}{K_p}$, $T=0.00625$s

$$p(k)=p(k-1)+\Delta p(k) \tag{14}$$

分组丢包概率

$$p(k)=\begin{cases}0, & c(k)<0\\ c(k), & 0\le c(k)\le 1\\ 1, & c(k)>1\end{cases} \tag{15}$$

3 PSO 算法与 PID 参数优化

PSO 算法只需很少的代码和参数,但在各种问题的求解与应用中却展现了它的特点和魅力. PSO 算法为人们提供了如下一种思路:使智能出现而不是努力强迫它;模拟自然而不是力图控制它;寻求使事情简化而不是让它复杂.

与进化算法比较,PSO 算法是一种更高效的并行搜索算法,它保留了基于种群的全局搜索策略,但是其采用的速度-位移模型操作简单,避免了像遗传算法中复杂的遗传操作. 它特有的记忆使其可以动态跟踪当前的搜索情况调整其搜索策略. PSO 算法基本能较快地达到全局最优值,PSO 算法基本不受问题峰数增加的影响,受问题维数的影响也很小.

与演化规划相比,PSO 算法执行一种有"意识(conscience)"的变异. 理论上,演化规划具有更多的机会在优化点附近开发,而 PSO 算法则有更多的机会更快地飞到更好解的区域.

根据 AQM 控制系统,考虑系统期望性能指标:调节时间、上升时间、超调量、静态误差、正弦跟踪误差. 运用 PSO 算法在给定的参数空间搜索,定义参数搜索空间如表 1 所示.

表 1 控制器参数搜索空间 Ω

控制器参数	最小值	最大值
k_p	0	1000
k_i	0	1000
k_d	0	1000

根据 AQM 控制系统的动静态性能在稳快准三方面的要求,定义一个如下的性能指标函数,以便在给定的参数搜索空间 Ω 上运用 PSO 算法搜索最优或次优的满足期望性能指标的 PID 控制器参数.

$$\min_{(k_p,k_i,k_d)}W(k_p,k_i,k_d)=e^{-\beta}(\sigma+e_s^2+e_{\sin})+(1-e^{-\beta})(t_s+t_r) \tag{16}$$

其中,σ为超调量; e_s 为静态误差; $e_{\sin}$为正弦跟踪误差(e_s, $e_{\sin}$的单位为数据包(pockets)); t_s 为调节时间; t_r 为上升时间(t_s, t_r 的单位为秒(s)); β为权系统,取合

适的 β 值可以调整调节时间和上升时间与超调量和静态误差、正弦跟踪误差之间的权重. σ 体现控制系统的平稳性, e_s、$e_{\sin}$ 为系统静态、动态误差, t_s、t_r 体现系统的快速性,性能指标函数定义的内涵在既保证系统的平稳性,还要满足系统的快速性和较小的稳态误差.

PSO 搜索算法步骤[13]如下:

第 1 步　初始化粒子群. 粒子个数 $n = 50$,粒子规模 $n \times 3$,每一个粒子的成员变量为 k_p, k_I 和 k_d,定义 t 为迭代序号. 在 Ω 内随机产生搜索点位置 $k_{j,g}^{(t)}$ 和对应的速度 $\upsilon_{j,g}^{(t)}$,计算对应的粒子个体极值和全局极值. 记录个体极值点位置 $pbest_{j,g}$ 和全局极值点位置 $gbest_g$, $k_g^{\min} \le k_{j,g}^{(k)} \le k_g^{\max}$, $\upsilon_g^{\min} \le \upsilon_{j,g}^{(t)} \le \upsilon_g^{\max}$, $\upsilon_g^{\min} = -\upsilon_g^{max}$, $j = 1,2,\cdots,n$, $g = 1,2,3$. 若 $g = 1$, $k_g^{\min} = k_p^{\min}$, $k_g^{\max} = k_p^{\max}$, $\upsilon_1^{\max} = \upsilon_{k_p}^{\max} = (k_p^{\max} - k_p^{\min})/2$, $\upsilon_{j,g}^{(t)}$ 表示第 t 次迭代第 j 号粒子对应 k_p 参数的速度, $k_{j,g}^{(t)}$ 表示第 t 次迭代第 j 号粒子对应 k_p 参数的位置;若 $g = 2$, $k_g^{\min} = k_i^{\min}$, $k_g^{\max} = k_i^{\max}$, $\upsilon_2^{\max} = \upsilon_{k_i}^{\max} = (k_{k_i}^{max} - k_i^{min})/2$, $\upsilon_{j,g}^{(t)}$ 表示第 t 次迭代第 j 号粒子对应 k_i 参数的速度, $k_{j,g}^{(t)}$ 表示第 t 次迭代第 j 号粒子对应 k_i 参数的位置;若 $g = 3$, $k_g^{\min} = k_d^{\min}$, $k_g^{\max} = k_d^{max}$, $\upsilon_3^{\max} = \upsilon_{k_d}^{\max} = (k_{k_d}^{\max} - k_d^{\min})/2$, $\upsilon_{j,g}^{(t)}$ 表示第 t 次迭代第 j 号粒子对应 k_d 参数的速度, $k_{j,g}^{(t)}$ 表示第 t 次迭代第 j 号粒子对应 k_d 参数的位置;转第 6 步.

第 2 步　计算每一个粒子对应系统的性能指标值 σ、e_s、$e_{\sin}$、t_s、t_r.

第 3 步　计算和每一个粒子对应的适应度函数 $W(k_p, k_i, k_d)$ 值.

第 4 步　对每个微粒,将其适应值与其经历过的最好位置作比较,若粒子的适应度值小于该粒子当前的个体极值,则将 $pbest_{j,g}$ 设置为该粒子的位置,并更新个体极值.

第 5 步　对每个微粒,将其适应值与全局经历过的最好位置作比较,若粒子中的个体极值优于当前的全局极值,则将 $gbest_g$ 设置为该粒子的位置,记录该粒子序号,并更新全局极值.

第 6 步　令
$$\upsilon_{j,g}^{(t+1)} = \tilde{\omega}\upsilon_{j,g}^{(t)} + c_1^* rand()^* (pbest_{j,g} - k_{j,g}^{(t)}) + c_2^* Rand()^* (gbest_g - k_{j,g}^{(t)}) \quad (17)$$
按式(17)更新粒子的速度.

其中, $\tilde{\omega} = \tilde{\omega}_{\max} - iter \cdot (\tilde{\omega}_{\max} - \tilde{\omega}_{\min})/itet_{\max}$, $\tilde{\omega}$ 为惯性权重,取 $\tilde{\omega}_{\max} = 0.9$, $\tilde{\omega}_{\min} = 0.4$, $iter$ 为迭代次数, $iter_{\max}$ 为最大迭代次数; c_1^*, c_2^* 为加速系数,一般取 $c_1^* = c_2^* = 2$; $rand()^*$, $Rand()^*$ 为 0 到 1 之间的随机数.

第 7 步　若 $\upsilon_{j,g}^{+1} > \upsilon_g^{\max}$,则 $\upsilon_{j,g}^{(t+1)} = \upsilon_g^{\max}$;若 $\upsilon_{j,g}^{(t+1)} < \upsilon_g^{\min}$,则 $\upsilon_{j,g}^{(t+1)} = \upsilon_g^{\min}$

第 8 步　令
$$\begin{cases} k_{j,g}^{(t+1)} = k_{j,g}^{(t)} + \upsilon_{j,g}^{(t+1)} \\ k_g^{\min} \le k_{j,g}^{(t+1)} \le k_g^{\max} \end{cases} \quad (18)$$
按式(18)更新粒子的位置.

第 9 步　若达到程序运行最大迭代次数 $iter_{\max}$,则转第 10 步,否则转第 2 步.

第 10 步　对应的粒子位置为最优粒子位置,即为所求最优 PID 控制器参数.

取 $\beta = 0.5$,最大迭代次数 $iter_{\max} = 50$,100 分别运行搜索程序,求得的结果如表 2 所示.

表 2　AQM 系统 PID 控制器参数及对应系统性能指标

迭代次数	β	k_p/10^{-8}	k_i/10^{-8}	k_d/10^{-8}	σ/%	e_s/pac	$e_{\sin}$/pac	t_r/10^{-5}s	t_r/10^{-5}s	$W(k_p, k_i, k_d)$
50	0.5	258.39	2982.91	3.56	0.7526	0	1.4628	4.7	2.8	1.3439
100	0.5	248.28	2890.30	3.40	0.7428	0	1.4312	4.5	2.7	1.3187

4　仿真研究

运用 *NS*2 网络仿真器验证本算法性能. 网络拓扑结构如图 6 所示,仿真实验结果与 RED 和 PI 两种典型算法进行比较.

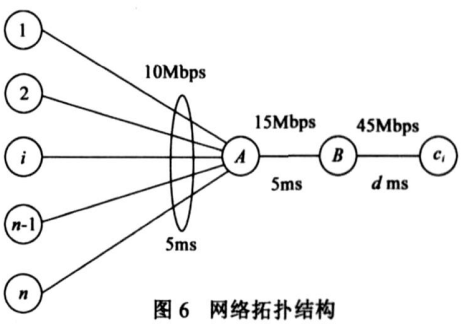

图 6　网络拓扑结构

节点 A 和节点 B 之间的瓶颈链路容量 15Mbps,延时 5ms. n 个持久性的 FTP 业务源与节点 A 之间的链路容量均为 10Mbps,通常情况之下延时 5ms,节点 B 和节点 C 之间的时延为 $d$ms. RED 高低门限值分别为 100packets 和 200packets, PI 和 PID 的队列长度的期望值为 150packets;各节点缓存大小均为 300packets.

实验 1　考察大时滞对算法性能的影响. n 取 60, 时延 d 取 220ms,所有 FTP 业务源均在 0 时刻启动. 瓶颈链路的容量为 15Mbps,RTT 时间约为 0.6s,主要包括传播时延、排队时延等. 采用前述方法,取最大迭代次数 $iter_{\max}$ 为 100,可求得 PID 控制器参数为: $K_p = 2.681 \times 10^{-5}$, $K_i = 4.632 \times 10^{-6}$, $K_d = 8.799 \times 10^{-6}$,实验仿真结果如图 7(*a*)、(*b*)、(*c*)所示.

从实验结果可以看出,RED 在大时滞中出现了持续震荡,PI 抖动厉害,调整到稳定状态需要的时间较长,丢包率较高,相比之下,基于 PSO 的 PID 算法具有较快的响应速度,综合性能较好. 各算法性能比较如表 3 所示,其中 σ 为超调量, t_s 为调节时间, e_{ss} 为稳态误差.

表 3 大时滞条件下各算法性能比较

性能指标 / 算法	σ %	t_s / s	e_{ss} / packets
RED	趋向 ∞	∞	系统不稳定,不求 e_{ss} 值
PI	100	35	10
PID(PSO)	5	5	2

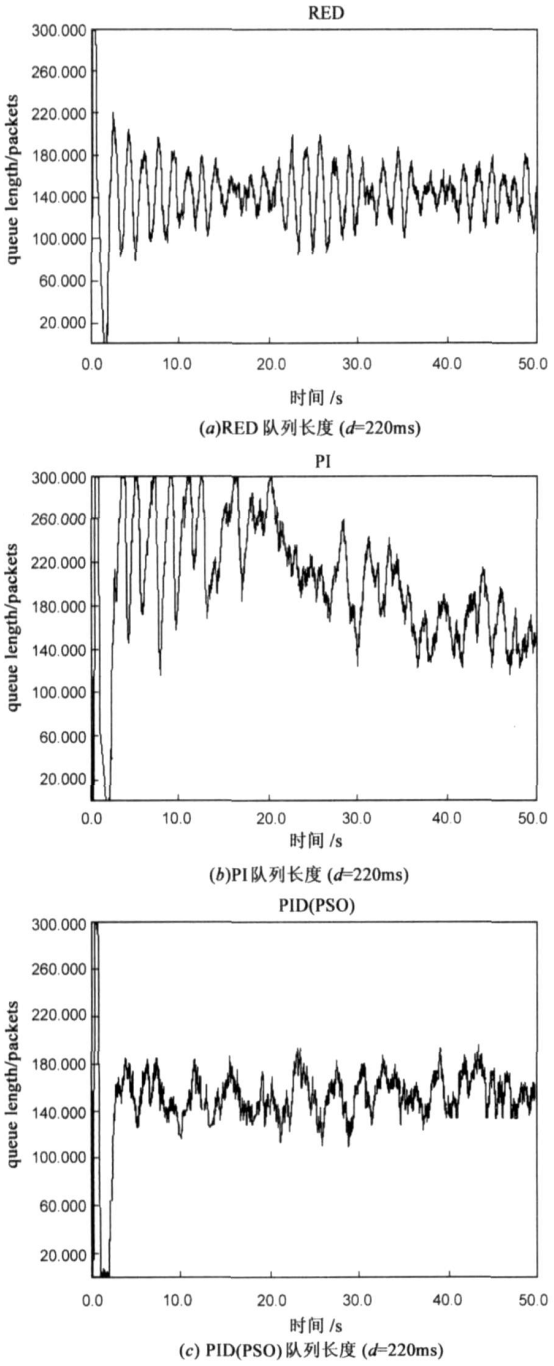

图 7

实验 2 考察突发业务流的冲击对算法的影响, n 取 70,时延 d 取 220ms,有 60 个 FTP 业务源均在 0s 时刻启动,还有 10 个在 15s 时刻启动,有 60 个 FTP 业务源均在 0 时刻启动,还有 10 个在 15s 时刻启动,发送 100k 字节后停止. 仿真结果如图 8(a)、(b)、(c)所示. 由图看出,当引入突发业务流时,RED、PI 影响最大,队列长度有所上升,而这些突发业务量终止时,其队列有所下

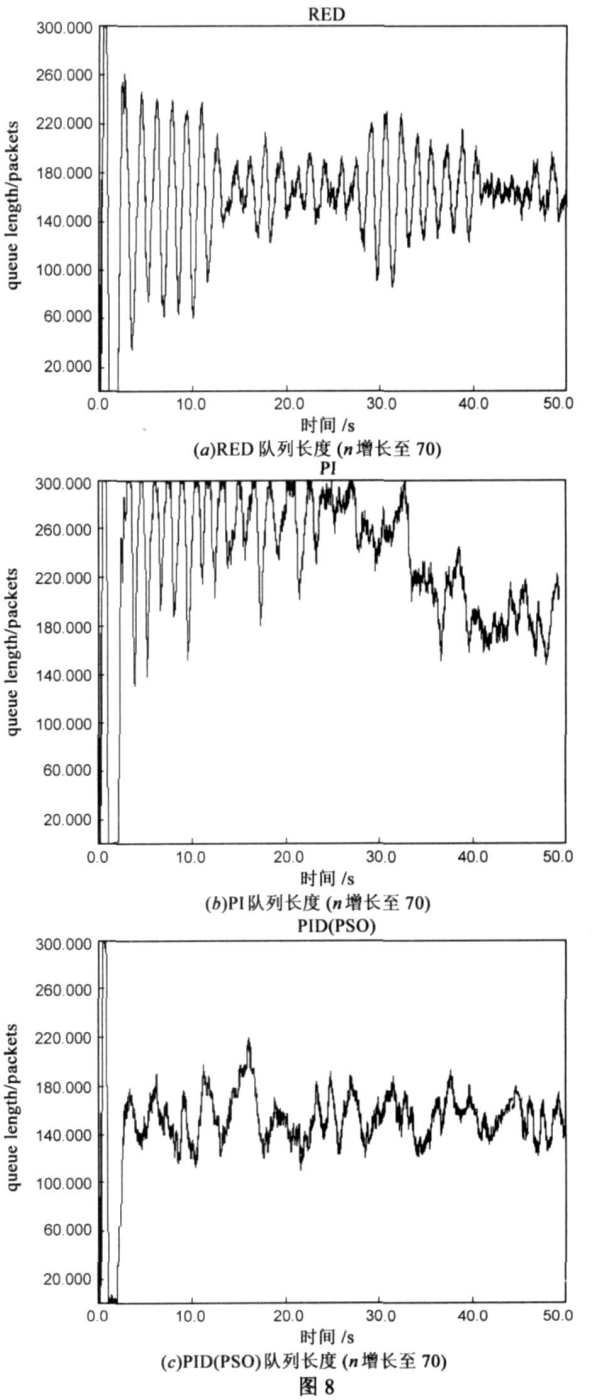

图 8

表 4 突发业务流的冲击对各算法性能影响比较

性能指标 / 算法	σ %	t_s / s	e_{ss} / packets
RED	趋向 ∞	40	系统不稳定,不求 e_{ss} 值
PI	100	35	20
PID(PSO)	5	4	3

降,出现较大振荡,相比之下,基于 PSO 的 PID 算法体现了较强的抗干扰能力,性能较好. 各算法性能比较如表 4 所示.

5 结论

本文推导了基于流体流理论的网络简化模型,基于该模型将 PID 控制器应用于网络 AQM 控制系统中,将集群智能中的改进粒子群优化算法(PSO)对 PID 控制器参数进行组合优化. 定义了一个综合调节时间、上升时间、超调量、系统静态误差、正弦跟踪误差等动静态性能指标函数,在给定的空间迅速搜索得使性能指标优化函数极小化的一组 PID 控制器参数. 仿真结果表明,基于改进粒子群优化(PSO)的 PID 控制算法具有较好的综合性能,比 RED、PI 算法更合适于 AQM 控制,表现为平均队列长度更趋于期望值;超调量更小;调节时间更短;队列长度的抖动更小;自适应能力更强. 在大时滞和突发业务流的冲击两种情况下,该方法设计的控制器超调量均小于 5 %,调节时间分别小于 5 秒、4 秒,稳态误差分别小于 2 个数据包和 3 个数据包.

参考文献:

[1] M Christiansen, K Jeffay, D Ott, F D Smith. Turing RED for eb traffic[J]. ACM Computer Communication Review, 2000, 30 (4): 139 - 150.

[2] W Feng, D Kandlur, D Saha, K Shin. A Self-Configuration RED gateway[A]. Proceedings of the INFOCOM'99[C]. New York: IEEE Computer Society, 1999. 1320 - 1328.

[3] T J Ott, T V Lakshman, L H Wong. SRED: stabilized RED [A]. Proceedings of the INFOCOM'99[C]. New York: IEEE Computer Society, 1999. 1346 - 1355.

[4] S Athuraliya, S Low, V H Li, Q H Yin. REM: Active queue management[J]. IEEE Network, 2001, 15(3): 48 - 53.

[5] V Misra, W B Gong, D Towsley. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED[A]. Proc ACM/ SIGCOMM[C]. Stockholm, 2000. 151 - 160.

[6] C V Hollot, V Misra, T D owsley, et al. A control theoretic analysis of RED [A]. Proc IEEE INFOCOM [C]. Alaska, USA, 2001. 1510 - 1519.

[7] C V Hollot, V Misra, D Towsley, et al. On designing improved controllers for AQM routers supporting TCP flows [A]. Proc IEEE INFOCOM[C]. Alaska, USA, 2001. 1726 - 1734.

[8] 陆锦军,王执铨. 基于速度控制的 API 网络拥塞控制策略[J]. 计算机应用, 2006, 26(5): 1137 - 1143.
Lu Jin-jun, Wang Zhi-quan. API network congestion contral scheme based on velocity control[J]. Journal of Computer Applications, 2006, 26(5): 1137 - 1143. (in Chinese)

[9] J Kennedy, R C Eberhart, Particle swarm optimization[A]. Proc of IEEE int'l Conf Nerual Networks[C]. Los Alamitos, CA: IEEE Computer Society Press, 1995. 1942 - 1948.

[10] R C Eberhart, J Kennedy. A new optimizer using particle swarm theory[A]. : Proc of the 6th Int'l Symp on Micro Machine and Human Science. Piscataway[C]. HJ: IEEE Service Center, 1995. 39 - 43.

[11] 冯奇峰,李言. 改进粒子群优化算法在工程优化问题中的应用研究[J]. 仪器仪表学报, 2005, 26(9): 984 - 990.
Feng Qi-feng, Li Yan. Research on the Application of IPSO in Engineering Optimization Problem[J]. Chinese Journal of Scientific Instrument, 2005, 26(9): 984 - 990. (in Chinese)

[12] 周殊,潘炜,罗斌,张伟利,丁莹. 一种基于粒子群优化方法的改进量子遗传算法及应用[J]. 电子学报, 2006, 34 (5): 897 - 901.
Zhou Shu, Pan Wei, Luo Bin, Zhang Wei-li, DING Ying. A novel quantum genetic algorithm based on particle swarm optimization method and its application[J]. Acta Electronica Sinica, 2006, 34(5): 897 - 901. (in Chinese)

[13] 李银伢,盛安东,王远钢. 基于粒子群优化的伺服系统比例积分微分控制器设计方法[J]. 兵工学报, 2006, 27(2): 202 - 205.
Li Yin-ya, Sheng An-dong, Wang Yuan-gang. A design method based on particle swarm optimization for PID controller of a servo system [J]. Chinese Acta Armamentarii, 2006, 27(2): 202 - 205. (in Chinese)

[14] 曾建湖,崔志华. 一种保证全局收敛的 PSO 算法[J]. 计算机机研究与发展, 2004, 4(8): 1334 - 1338.
Zeng Jian-Chao, Cui Zhi-Hua. A guaranteed global convergence particle swarm optimizer[J]. Journal of Computer Research and Development, 2004, 4(8): 1334 - 1338. (in Chinese)

作者简介:

陆锦军　男,副教授,主要研究方向为网络系统的拥塞控制,智能控制理论与应用.
E-mail: Ljj @mail. ntvc. edu. cn

王执铨　男,教授,博士生导师,目前感兴趣的研究方向为网络系统的拥塞控制,混沌控制,鲁棒控制,容错控制,信息系统的安全理论与技术等等.