第 9 卷第 1 期 2011 年 3 月
1672-6553/2011/09(1)/054-4

动力学与控制学报
JOURNAL OF DYNAMICS AND CONTROL

Vol. 9 No. 1
Mar. 2011

一种基于神经元强化学习的网络拥塞控制方法*

周川　狄东杰　陈庆伟　郭毓

（南京理工大学自动化学院，南京　210094）

摘要　提出了一种基于神经元强化学习（Neuron-based Reinforcement Learning, NRL）的自适应 AQM 算法，采用链路速率和队列长度作为拥塞指示，可根据网络环境的变化在线自动调整神经元参数，从而保持良好的队列长度稳定性和对网络负载波动的鲁棒性. 该算法结构简单、易于实现，且不依赖对象的模型. 仿真结果表明，该算法尤其适合于解决复杂不确定性网络的拥塞控制问题，并具有更好的队列稳定性和鲁棒性.

关键词　拥塞控制，　主动队列管理（AQM），　神经元，　强化学习

引 言

随着 Internet 的飞速发展，网络业务量不断增长，人们对网络服务性能（QoS）提出了更高的要求，仅依靠 TCP 的流控制（Flow Control）机制和源端控制算法无法避免拥塞的发生，因此在位于网络中间节点的路由器处进行拥塞控制，即主动预见式地丢弃分组可有效防止拥塞的发生，并可有效克服传统被动式丢尾（Drop Tail）策略下的满队列、死锁和全局同步问题，因而其研究已成为网络与通讯和控制领域的研究热点[1][2][3][4][5][6].

目前网络拥塞控制中的 AQM 算法主要可分为启发式算法、基于控制理论的算法和基于优化的算法等. Floyd 在文献[2]中提出了 RED 算法，采用平均队列长度作为拥塞指示，避免了全局同步和死锁现象，但其存在对参数设置敏感的缺点. PI 算法[9]可增强了 AQM 系统的稳态性能，但存在瞬态特性不理想、过分依赖缓存空间大小等问题. 而 PID 算法[9]可通过引入微分作用改善瞬态性能，缩短 AQM 系统的调节时间，但缺乏对动态网络环境的自适应机制，尤其当网络负载波动的情况下则难以保证队列稳定性且对参数变化的鲁棒性不足. REM[4]算法以队列长度和链路速率作为拥塞指示，能够及时检测到拥塞并对数据包进行丢弃/标记，但 REM 对网络参数的变化较为敏感. 因此，针对复杂时变的网络环境，研究具有适应性和鲁棒性的 AQM 算法已成为一个重要的理论和技术问题.

近年来，鉴于智能控制在解决复杂网络系统中的非线性、时变和不确定性等问题具有优势，且智能 AQM 算法具有对模型失配和参数变化的鲁棒性及适应性等优点，因此基于智能控制技术的 AQM 算法研究将成为一个新的热点. 文献[5]提出基于队列误差和队列误差变化率作为拥塞指示的模糊 AQM 控制算法，并结合 PI 控制器进行补偿来保持队列稳定在期望值，但该方法的模糊规则获取具有主观性且难以在线更新，因而对网络参数变化的适应能力有限，且动态响应存在滞后性. 本文在综合启发式算法和神经元强化学习的基础上，以队列长度和链路速率作为网络拥塞指示量，引入队列长度与期望队列长度以及链路速率与链路容量这二个误差量作为神经元的输入，并通过强化学习算法在线调整神经元的权值参数. 该算法具有更快的队列响应，同时又保持较小的队列延时，并具有良好的队列稳定性和对网络负载波动的鲁棒性.

1　问题描述

文献[6]提出采用流体理论和随机非线性微分方程来建立 TCP 动态模型，若忽略 TCP 的超时机制，则可用如下一组非线性微分方程描述：

$$\frac{dW(t)}{dt}=\frac{1}{R(t)}-\frac{W(t)}{2}\frac{W(t-R(t))}{R(t-R(t))}p(t-R(t)) \tag{1a}$$

$$\frac{dq(t)}{dt}=\frac{W(t)}{R(t)}N(t)-C \tag{1b}$$

2010-10-20 收到第 1 稿，2010-11-21 收到修改稿.
* 江苏省自然科学基金（BK2007206），南京理工大学自主科研专项计划（2010GJPY066）和南京市留学回国启动基金资助项目

其中,W 为 TCP 窗口大小(包);q 为队列长度(包);$R(t)$为回路往返时间,且 $R(t)=T_p+q(t)/C$;T_p 为传输延迟;C 为链路容量(包/秒);$N(t)$为 TCP 连接数;$p(t)$为分组丢弃/标记概率,且 $p(t)\in[0,1]$. 队列长度 q 和窗口大小 W 均为正的有界量,即 $q\in[0,\bar{q}]$,$W(t)\in[0,\bar{W}]$,其中 $\bar{q}$ 和 $\bar{W}$ 分别为缓存容量和最大窗口尺寸.

若假定 $R(t)=R_0$ 为常量,且每个源端的窗口值具有相同的变化特性时,链路输入速率 $x(t)$与窗口值 $W(t)$满足:$x(t)=W(t)N/R_0$,则式(1)可表示为:

$$\dot{x}(t)=\frac{N}{R_0^2}-\frac{x(t)x(t-R_0)}{2N}p(t-R_0) \tag{2a}$$

$$\dot{q}=x(t)-C \tag{2b}$$

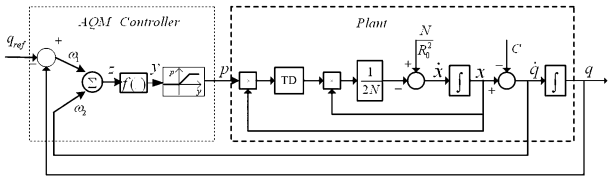

图 1　基于神经元强化学习的 AQM 控制系统

Fig. 1　AQM control system based on neuron reinforcement learning

基于神经元强化学习的 AQM 控制系统结构如图 1 所示,其中系统的输入为路由器期望队列长度 q_{ref},输出为实际队列长度 q,其中 TD 为延迟算子,且延迟大小为 R_0. 受控对象为 TCP 源端的速率调节与队列变化模型由非线性时延微分方程(2)式表示,AQM 控制器的输入分别为队列长度与期望值的偏差量 $q(t)-q_{ref}$及其速率变化量 $x(t)-C$. 主动队列算法主要通过神经元强化学习来决策出丢弃/标记概率 $p(t)$,使得中间节点的队列长度 $q(t)$尽可能保持在期望的队列长度 q_{ref}附近.

2　基于神经元强化学习的自适应 AQM 算法及其实现

本文提出的基于神经元强化学习的 AQM 算法采用路由器队列长度与输入速率 $x(t)$作为拥塞指示,针对时变的网络环境,通过强化学习算法在线调整神经元的连接权值动态决策出丢弃/标记概率 $p(t)$,对网络拥塞及时做出响应,从而对网络的时变参数和不确定性具有较强的适应性和鲁棒性. 图 1 所示的 AQM 控制器由单神经元实现,其输入分别为队列长度误差 $q(t)-q_{ref}$与速率的误差 $x(t)-C$,记 $e(t)=[q(t)-q_{ref},x(t)-C]^T$ 为输入向量,神经元输入权值为 $\omega(t)=[\omega_1(t),\omega_2(t)]^T$,神经元的净输入为 $z(t)$,则 $z(t)$可表示为:

$$z(t)=\omega_1(t)[q(t)-q_{ref}]+\omega_2(t)[x(t)-C]=\omega^T(t)e(t) \tag{3}$$

神经元的输出为 $y(t)$,若选取神经元激活函数为:

$$y(t)=f(z(t))=\frac{1-e^{-z(t)}}{1+e^{-z(t)}} \tag{4}$$

式(4)中 S 型激活函数将输入 $z(t)$映射到 $y(t)\in[-1,1]$,经过输出限幅函数可决策出丢弃/标记概率 $p(t)$

$$\begin{cases}0 & y(t)<0\\ y(t) & 0\leqslant y(t)\leqslant 1\\ 1 & y(t)>1\end{cases} \tag{5}$$

在强化学习中系统的目标是在最优控制策略 $\pi^*:S\rightarrow A$(其中 S 是状态集,A 是动作集)引导下,得到最大的累积奖赏值[7]. 累积奖赏值可表示为:

$$R_s(t)=r(t+1)+\gamma r(t+2)+\gamma^2 r(t+3)+\cdots=\sum_{k=0}^{\infty}\gamma^k r(t+k+1)=r(t+1)+\gamma R_s(t+1) \tag{6}$$

式(6)表示系统针对环境在某个策略 π 指导下的一次学习中,从 s 状态往后所获得的所有奖赏的累计折扣和,其中 γ 为折扣因子,$r(t+1)$为瞬时奖赏函数,可选取瞬时奖赏函数

$$r(t+1)=-\theta_1(q(t)-q_{ref})^2-\theta_2(x(t)-c)^2 \tag{7}$$

其中 θ_1,θ_2 为相应的权重.

由于环境是不确定的,系统在某个策略 π 指导下的每一次学习环境中所得到的 $R_s(t)$有可能是不同的. 因此在 s 状态下的累积奖赏值函数要考虑不同学习环境中所有函数的数学期望,故系统在 s 状态下,遵循某个 π 策略时,所能获得的累计折扣的数学期望如下式:

$$V^{\pi}(t)=E_{\pi}\{R_s(t)\mid s(t)=s\}=E_{\pi}\{r(t+1)+\gamma\sum_{k=0}^{\infty}\gamma^k r(t+k+2)\mid s(t)=s\}=E_{\pi}\{r(t+1)+\gamma V^{\pi}(t+1)\mid s(t)=s\} \tag{8}$$

如果在最优策略 π^* 下得到最大的累积奖赏值 V^* 为期望的目标值. $V(t)$为从 $s(t)$状态往后所得到的累计奖赏的估计值,神经元学习的目标就是使 $V(t)$充分逼近最优策略下期望值 V^*,且神经元在线学习可采用瞬时差分(Temporal Difference,TD($\lambda=0$))算法[8]. 构造一个 V 的估计函数为 $V(t)=z(t)=\omega^T(t)e(t)$. 定义性能函数为

$$E(t)=\frac{1}{2}[V^*-V(t)]^2 \tag{9}$$

神经元权值调整的目标是使 $E(t)$ 取得最小值. 权值调整采用如下梯度学习算法

$$\omega(t+1)=\omega(t)-\alpha\frac{\partial E(t)}{\partial\omega(t)} \tag{10}$$

其中 $\frac{\partial E(t)}{\partial\omega(t)}=-[V^*-V(t)]\nabla_{\omega(t)}V(t)$, $\alpha>0$ 为学习速率,则

$$\omega(t+1)=\omega(t)+\alpha[V^*-V(t)]e(t) \tag{11}$$

由式(8)可知最优策略 π^* 下得到最大的累积奖赏值数学期望为 $V^*=E\{r(t+1)+\gamma V^*(t+1)\mid s(t)=s\}$,且由于 $V^*(t+1)$ 未知,可用估计值 $V(t+1)$ 进行替代,并将其带入(11)式,可得神经元权值的调整规则为

$$\omega(t+1)=\omega(t)+\alpha[r(t+1)+\gamma V(t+1)-V(t)]e(t) \tag{12}$$

基于神经元强化学习的 AQM 算法的伪代码为图 2 所示:

```
q: current queue length   q_ref: desired queue length   x: current rate
C: link capacity   θ1,θ2: weights of penalty function   α: learing rate
γ: discount factor   e=[e1,e2]: neuron input   V: state value function
ω1,ω2: the adjustable weights of neuron

Calculate_p()
{ e1=q-q_ref; e1=x-C;
  r=-θ1*e1^2-θ2*e2^2;
  ω1=ω1+α(r+γ*V-V_old)*e1;
  ω2=ω2+α(r+γ*V-V_old)*e2;
  V_old=V;
  V=ω1*e1+ω2*e2;
  p=2/(1+e^V)-1;}

Enqueue() /*each packet arrival*/
  { random=uniformRandom(0,1);
    if(random<p)
      { Enqueue the packet;}
    else
      { drop the packet; }
  }
```

图 2 AQM 算法的伪代码

Fig. 2 Pseudo – code of AQM algorithm

3 仿真实验与性能分析

采用网络仿真软件 NS – 2 来验证神经元强化学习算法的性能,并将该方法与标准的 REM 算法,RED 算法,PI 算法进行比较. 仿真所采用的网络拓扑结构如图 3 所示,瓶颈链路位于 R1 和 R2 之间,其链路容量为 15Mbps,时延为 20ms,该瓶颈链路被个 TCP 流共享,分组大小为 500bytes,缓冲队列长度为 400 个包. 两端分别为发送端和接收端,发送节点与 R1 之间的链路容量以及接收节点与 R2 之间的链路容量均为 10Mbps,链路延时均为 2ms. 在仿真中期望队列长度设置为 q_{ref} = packets,且仿真时间为 100s. 为了实现负载波动情况下的仿真,在 0s,20s,40s,60s,80s 分别在发送端依次启动 50 个 TCP 流. 神经元强化学习(NRL)算法的参数选为: $\alpha=0.001$, $\gamma=0.98$, $\theta_1=10$, $\theta_2=0.02$, $q_{ref}=200$.

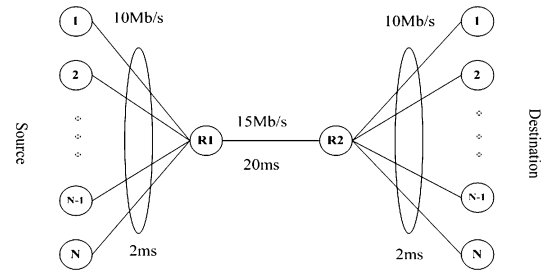

图 3 仿真实验网络拓扑结构

Fig. 3 Network topology for simulation

仿真实验结果如图 4 所示,分别给出了 NRL、REM、RED 和 PI 四种 AQM 算法的队列长度响应. 由图可知,NRL 算法能很好的维持队列长度在期望值附近,且波动较小,响应时间最短,并表现出较好的鲁棒性;REM 算法与 RED 算法在负载发生变化时波动剧烈,且响应时间长;PI 算法则与期望队列程度有较大偏差且稳定性较差. 仿真实验表明在负载变化情况下 NRL 算法在队列稳定性和响应速度等方面优于其他三种算法,对网络负载的波动具有很强的鲁棒性.

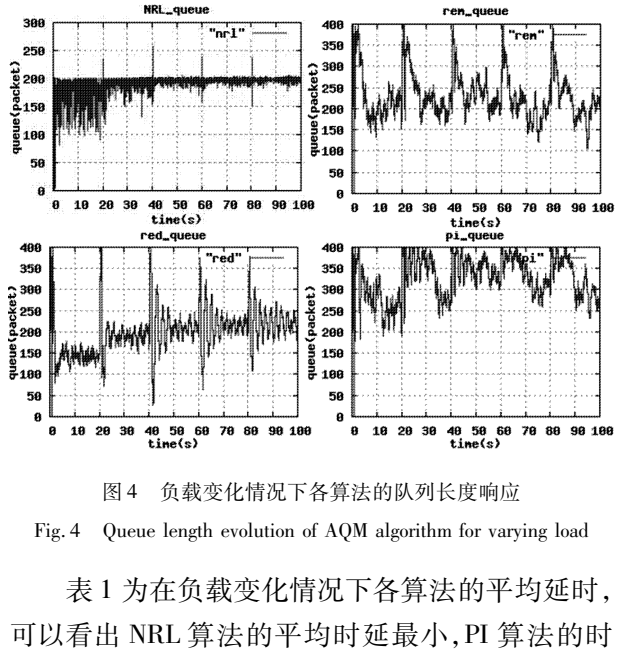

图 4 负载变化情况下各算法的队列长度响应

Fig. 4 Queue length evolution of AQM algorithm for varying load

表 1 为在负载变化情况下各算法的平均延时,可以看出 NRL 算法的平均时延最小,PI 算法的时延最大.

表 1 负载变化时各算法的平均时延

Table 1 Average delay of AQM algorithm for varying load

Algorithm	NRL	NRL	RED	PI
Average delay(ms)	105.121	126.812	110.074	180.262

综上可见,本文的 NRL 算法在队列稳定性、鲁棒性和动态响应速度等方面要优于其他三类算法,其原因在于增加了输入速率作为拥塞指示特征,能迅速对网络拥塞做出响应,同时由于采用神经元强化学习算法,能够根据复杂未知的网络环境(如负载的变化等因素)来在线调节 AQM 控制器的参数,从而具有较强的鲁棒性和适应性.

4　结论

本文针对 TCP 网络模型的拥塞控制问题,提出了一种基于神经元强化学习的自适应主动队列管理算法,该算法以队列长度与输入速率作为拥塞指示反馈信号,神经元 AQM 控制器采用了强化学习算法能够针对复杂网络环境的参数变化和不确定性进行参数在线调整,且该算法结构简单,易于实现. 仿真结果验证了该算法能使队列稳定性好且动态响应速度快,并具有较强的适应性和鲁棒性.

参　考　文　献

1　Floyd S, Fall K. Promoting the use of End-to-End congestion control in the Internet. *IEEE/ACM Transactions on Networking*, 1999, 7(4):458 ~ 472

2　Floyd S, V Jacobson. Random early detection gateway for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1993, 1(4):397 ~ 413

3　Athuraliya S, Steven H Low, et al. REM: active queue management. *IEEE/ACM Transactions on Networking*, 2001, 15(3): 48 ~ 53

4　Hollot C V, Misra V, Towsley D, Gong W. On designing improved controllers for AQM routers supporting TCP flows. *Proc. IEEE INFOCOM*, 2001, 3: 1726 ~ 1734

5　Aoul Y H, et al. A fuzzy logic-based AQM for real-time traffic over Internet. *Computer Networks*, 2007, 16: 4617 ~ 4633

6　Misra V, Gong W, Towsley D. Fluid-based analysis of a network of AQM routers supporting TCP flows. *Proc. ACM/SIGCOMM*, 2000, 30(4):151 ~ 160

7　Sutton R, Barto AG. Reinforcement learning: An Introduction. MIT Press, 1998

8　Dayan P. The convergence of TD (λ) for general λ. *Machine Learning*, 1992, 8:341 ~ 362

9　Ren F Y, Wang F B, Ren Y, et al. PID controller for active queue management. *Journal of Electronics and Information Technology*, 2003, 25 (1):94 ~ 99

10　朱海峰,李伟,张林. 基于 BP 神经网络整定的 PID 控制. 动力学与控制学报,2005,3(4):93 ~ 96 (Zhu H F, Li W, Zhang L. PID control based on BP neural network adjusting. *Journal of Dynamics and Control*, 2005, 3(4): 93 ~ 96 (in Chinese))

A CONGESTION CONTROL SCHEME BASED ON NEURON REINFORCEMENT LEARNING*

Zhou Chuan　Di Dongjie　Chen Qingwei　Guo Yu

(*School of Automation*, *Nanjing University of Science & Technology*, *Nanjing*　210094, *China*)

Abstract　A novel adaptive AQM scheme based on Neuron Reinforcement Learning (NRL) was presented. This scheme uses queue length and link rate as congestion notification to determine an appropriate drop/mark probability, and the parameters of neuron can be adjusted online according to the time-varying network environment, so that the stability of queue dynamics and robustness for fluctuation of TCP loads are guaranteed. This scheme is easy to implement with simple structure, and it is independent of the system model to be controlled. Simulation results show that the proposed algorithm can be effective in solving congestion control problem for complex network with uncertainties, and it has better performance on stability and robustness.

Key words　congestion control,　Active Queue Management,　neuron,　reinforcement learning

Received 20 October 2010, revised 21 November 2010.

* This work was supported by the Natural Science Foundation of Jiangsu province (BK2007206), NUST Research Foundation (2010GJPY066) and Research Foundation for the ROCS of Nanjing