

# 自适应卡尔曼滤波的主动队列管理算法

闫 巧<sup>1</sup> 胡晓娟<sup>2</sup> 雷琼钰<sup>2</sup>

(深圳大学计算机与软件学院 深圳 518060)<sup>1</sup> (深圳大学信息工程学院 深圳 518060)<sup>2</sup>

**摘 要** PID 控制器通过微分环节加快了控制器的调节速度,但 PID 的参数是固定的,不能根据动态的网络自调整参数,故不能有效控制队列的稳定性。由于神经网络有自适应性,提出了一种自适应卡尔曼滤波的主动队列管理算法(adaptive-KF-AQM)。它结合卡尔曼滤波和神经网络方法,根据队列长度及其变化率来估计下一时刻的队列长度,使队列长度在期望值附近波动。仿真结果表明,该算法在队列稳定性、收敛速度、延时和链路利用率等方面都明显优于传统的 PID 算法。

**关键词** 卡尔曼滤波器,自适应,单神经元,神经网络,主动队列管理

**中图法分类号** TP393.01 **文献标识码** A

## New Adaptive Active Queue Management Algorithm with Kalman Filter

YAN Qiao<sup>1</sup> HU Xiao-juan<sup>2</sup> LEI Qiong-yu<sup>2</sup>

(College of Computer and Software, Shenzhen University, Shenzhen 518060, China)<sup>1</sup>

(College of Information Engineering, Shenzhen University, Shenzhen 518060, China)<sup>2</sup>

**Abstract** PID controller accelerates the regulation speed of the controller through differential factor. But the parameters of PID controller are fixed, they can't be adapted with dynamic network, so the stability of the queue can't be controlled effectively. A new adaptive active queue management(AQM) algorithm with Kalman filter was presented according to the adaptivity of the neural network. The new algorithm combines Kalman filter law with neural network, which has the merits of both. It can determinate future queue length based on queue lengths and some rates of change in the queue length. The results of simulation show that the new AQM algorithm is superior to the typical PID controller on the queue stability, time delay and link utilization.

**Keywords** Kalman filter, Adaptive, Neuron, Neural network, AQM

## 1 引言

随着互联网规模的扩大,拥塞已经成为一个十分重要的问题。为了解决这一问题,人们提出了主动队列管理(Active Queue Management, AQM)算法。AQM 作为路由器中的一项重要技术,主要是控制队列长度,通过随机丢弃数据包来通知源端发生拥塞。文献[1]提出了著名的随机早期检测算法(Random Early Detection, RED),该算法可使不同的 TCP 流拥塞响应异步化,从而解决了 TCP 全局同步问题。但是,在不同的网络环境下难以配置 RED 参数,而且 RED 参数对网络负载很敏感。

近年来,在主动队列管理方面提出了基于控制理论的分析 and 设计,许多研究者依据控制理论的方法来设计 AQM 控制器。文献[2]在非线性的 TCP/AQM 动态模型的基础上,利用线性系统和经典控制理论的方法设计了 PI 控制器。后来又出现了 PID<sup>[3]</sup>等控制器。为了克服 PID 控制器的弱点,文献[4]通过神经网络学习算法调节控制器的参数,并在此基础上调节丢包率,避免网络拥塞。为了克服网络系统模型的不

精确性和参数设置的局限性,文献[5]采用鲁棒控制理论设计、分析 AQM 机制,使用一个输出反馈来控制延迟抖动,减少模型不确定性的影响。文献[6]用现代控制理论中的最优滤波器和目标追踪方法来设计 AQM 控制器,以便在长期流和突发短期流时减小队列变化。

为了使队列长度收敛在期望值附近,各种算法都应选择合理的参数。但因网络的动态变化,导致参数的选择变得十分困难,固定的参数值无法满足网络的动态性。基于这个问题,提出了一种基于单神经网络的卡尔曼自适应滤波算法,即通过神经网络的不断学习,在线调整参数,从而对拥塞做出及时的响应。

## 2 PID 主动队列管理算法

PID<sup>[3]</sup>主动队列管理算法的主要思想是通过比较当前队列长度  $q$  与期望队列长度  $q_0$ ,并记录它们的差值  $e(t)$ ,然后将差值输入 PID 控制器中,得到当前数据包丢弃概率  $p$ 。算法的实现框架如图 1 所示。

收稿日期:2011-03-20 返修日期:2011-05-19

闫 巧(1972—),女,博士,教授,主要研究方向为网络安全, E-mail: yanq@szu.edu.cn; 胡晓娟(1986—),女,硕士生,主要研究方向为网络拥塞控制; 雷琼钰(1987—),女,硕士生,主要研究方向为网络拥塞控制、网络安全。

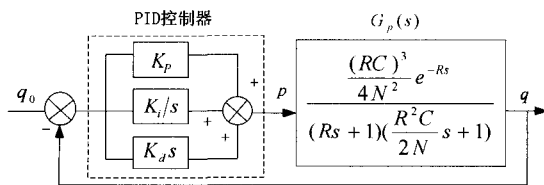


图1 PID控制器<sup>[3]</sup>

其中,  $K_p$ 、 $K_i$ 、 $K_d$  分别为控制器的比例、积分、微分系数,  $G_p(s)$  为被控对象的传递函数。

令  $t = kT$ ,  $T = \frac{1}{f_s}$ 。其中  $T$  为采样周期,  $f_s$  为采样频率,

对该过程进行离散化, 丢弃概率的表达式为<sup>[3]</sup>

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

增量形式为<sup>[3]</sup>

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

式中,  $T_i = \frac{K_p}{K_i}$ ,  $T_d = \frac{K_d}{K_p}$ 。

### 3 自适应卡尔曼滤波的主动队列管理算法

#### 3.1 卡尔曼滤波的主动队列管理算法(KF-AQM)

基于卡尔曼滤波的主动队列管理算法是由 Shane F. Cotter<sup>[6]</sup>等人提出来的。该算法的主要思想是通过当前队列长度、前一时刻队列长度和前一时刻的队列长度变化率来确定下一时刻的队列长度。依据卡尔曼滤波器原理, 先构造出一个离散时间随机系统, 令  $q_k$  是在  $kT$  时刻的队列长度,  $v_k$  是在  $kT$  时刻的队列长度变化率。而当前时刻的队列长度<sup>[6]</sup>和队列长度变化率<sup>[6]</sup>为

$$q_k = q_{k-1} + v_{k-1}T + \delta_{k-1} \quad (1)$$

$$v_k = v_{k-1} + \epsilon_{k-1} \quad (2)$$

令  $kT$  时刻的状态矢量<sup>[6]</sup>为  $x_k = [q_k \quad v_k]^T$ , 故上式可写

成<sup>[6]</sup>  $x_k = Ax_{k-1} + \xi_{k-1}$ ,  $A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$ ,  $\xi_{k-1} = [\delta_{k-1} \quad \epsilon_{k-1}]^T$ , 它是符合  $N(0, V_{k-1})$  分布的高斯噪声,  $T$  是采样时间。令输出

方程<sup>[6]</sup>为  $y_k = q_k + e_k$ , 其中  $y_k$  是在  $kT$  时刻缓冲器中总的数据包,  $e_k$  也为高斯噪声, 符合  $N(0, R_{k-1})$ , 同时输出方程也可以写成<sup>[6]</sup>  $y_k = Cx_k + e_k$ ,  $C = [1 \quad 0]$ 。

由此构造出的离散时间随机系统<sup>[6]</sup>为  $\begin{cases} x_k = Ax_{k-1} + \xi_{k-1} \\ y_k = Cx_k + e_k \end{cases}$ ,

根据卡尔曼滤波公式, 可得状态预报<sup>[6]</sup>

$$x_k^- = Ax_{k-1}, P_k^- = AP_{k-1}A^T + V_{k-1} \quad (3)$$

状态估计<sup>[6]</sup>为

$$K_k = P_k^- H^T S_k^{-1}$$

其中,

$$S_k = (HP_k^- H^T + R_k) \quad (4)$$

$$P_k = P_k^- - K_k S_k K_k^T \quad (5)$$

$$x_k = x_k^- + K_k (y_k - Hx_k^-) \quad (6)$$

最后假设缓冲器的容量为  $B$ , 此时主动队列管理算法的丢包率<sup>[6]</sup>为

$$p = \begin{cases} 0, & q_{k+1}^- < aB \\ \frac{q_{k+1}^- - aB}{B - aB}, & aB \leq q_{k+1}^- < B \\ 1, & q_{k+1}^- \geq B \end{cases} \quad (7)$$

式中,  $a$  为常数。

假设卡尔曼滤波算法中的状态矢量  $x$  的理论初值为  $[100, 1]$ , 采样时间为  $0.00625s$ , 可用 matlab 求出滤波增益矩阵  $K = [0.62426, 0.61611]^T$ , 得滤波效果显示图, 如图 2 所示。其中队列长度  $x_1$  的滤波值很快就接近理论值, 两条曲线几乎重合, 而队列长度变化率  $x_2$  的滤波值要经过 500 个时间间隔后才慢慢接近理论值。

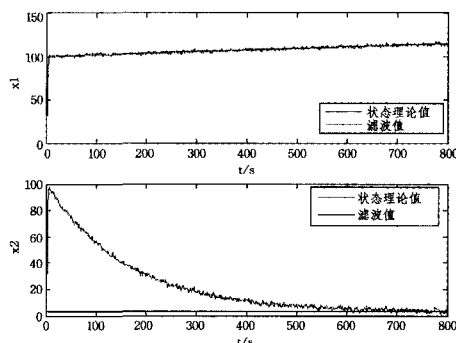


图2 滤波效果显示图

#### 3.2 基于单神经元的卡尔曼滤波主动队列管理算法(adaptive KF-AQM)

本文基于单神经元具有在线自学习能力, 并且结构简单、易于实现。将单神经元与卡尔曼滤波算法相结合, 可以提高卡尔曼滤波算法在复杂和时变非线性系统中的性能。单神经元的系统结构如图 3 所示<sup>[7]</sup>。通常将输入信号  $x_j$  乘以连接权重  $w_{ij}$ , 再通过阈值  $\theta_i$  后的结果作为激励函数  $f(x)$  的输入, 而  $f(x)$  的输出就是神经元的输出。根据需解决的问题不同, 选择不同的  $f(x)$ , 或为线性函数, 或为非线性函数。

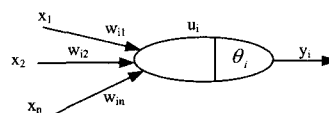


图3 单神经元结构图<sup>[7]</sup>

其中,  $u_i$  为神经元的内部结构,  $x_j$  为输入信号,  $w_{ij}$  为连接权重值,  $j=1, 2, \dots, \theta_i$  为阈值,  $y_i$  为神经元的输出。

在控制系统中定义了两个独立神经元: 一个用于估计队列长度, 一个用于估计队列长度的变化率。先看第一个用于估计队列长度的神经元, 其前向网络有 3 个输入, 分别为期望队列长度与实际队列长度的差值  $q_{ref} - q(k)$ 、当前时刻的队列长度变化率  $var$  和前一时刻的队列长度  $q(k-1)$ 。在任意采样时刻  $k$ , 其输入信号为

$$x_i' = \begin{cases} q(k-1), & i=1 \\ var(k), & i=2 \\ q_{ref} - q(k), & i=3 \end{cases} \quad (8)$$

根据卡尔曼滤波原理, 此神经元的输入为  $net' = \sum_{j=1}^3 w_{ij}' x_j'$ 。接着选择一个比例函数作为神经元的激励函数, 故该神经元的输出为

$$y_1 = f(net') = w_{11} q(k-1) + w_{12} var(k) + w_{13} (q_{ref} - q(k)) \quad (9)$$

对于第二个用于估计队列长度的变化率独立神经元,其前向网络有两个输入,分别为前一时刻的队列长度的变化率和随机误差。随机误差符合高斯分布,在任意采样时刻  $k$ ,其输入信号为

$$x_i'' = \begin{cases} \text{var}(k), & i=1 \\ e(k), & i=2 \end{cases} \quad (10)$$

根据卡尔曼滤波原理,此神经元的输入为  $net'' = \sum_{j=1}^2 w_{ij}'' x_j''$ 。同理选择一个比例函数作为该神经元的激励函数,故该神经元的输出为

$$y_2 = f(net'') = w_{21} \text{var}(k) + w_{22} e(k) \quad (11)$$

由于算法的主要目的是将队列长度收敛于期望队列长度,并使其在期望值附近波动,因此引入的性能指标函数为  $E(k) = \frac{1}{2} [q_{\text{ref}} - q(k)]^2$ 。而神经元训练和学习的目的是使  $E(k)$  值最小。

下面以估计队列长度的神经元为例,按梯度下降法来调节两神经元的权重值,而另一个估计队列长度变化率的神经元与之类似。经过  $n_0$  步训练和学习后,两神经元权重值的迭代方程为

$$w_{ij}(n_0+1) = w_{ij}(n_0) - \eta \frac{\partial E}{\partial w_{ij}} \quad (12)$$

在调节权重值的过程中,不仅要考虑误差在梯度上的作用,还要考虑在误差曲面上变化趋势的影响。为了避免在权值的调节过程中发生振荡、收敛速度慢以及神经网络陷入局部极小值,可利用附加动量法。该方法实质上考虑了前一次权值变化对本次权值变化的影响,从而保证权值的学习朝着误差曲面底部的平均方向修正,并根据梯度下降算法来获得新的权值。带有附加动量因子的权值修正公式为

$$w_{ij}(n_0+1) = w_{ij}(n_0) - \eta \frac{\partial E}{\partial w_{ij}} + \alpha(w_{ij}(n_0) - w_{ij}(n_0-1)) \quad (13)$$

式中,

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial q(k)} \frac{\partial q(k)}{\partial y} \frac{\partial y}{\partial net} \frac{\partial net}{\partial w_{ij}} \quad (14)$$

式中,  $\frac{\partial E}{\partial q(k)} = -(q_{\text{ref}} - q(k))$ ,  $\frac{\partial y}{\partial net} = 1$ ,  $\frac{\partial net}{\partial w} = 1$ 。  $\frac{\partial q(k)}{\partial y}$  绝对值大小将影响权值收敛的变化速度,但  $\frac{\partial q(k)}{\partial y}$  的正负却影响权值收敛的方向。对于神经网络权值的收敛,收敛方向是最重要的,收敛速度还可以通过其他因素来调节。所以采用  $q(k)$  与  $y(k)$  的相对变化量的符号函数近似代替:

$$\frac{\partial q(k)}{\partial y} = \text{sgn}\left(\frac{q(k) - q(k-1)}{y(k) - y(k-1)}\right) \quad (15)$$

最后将式(14)、式(15)带入式(13),得到修正后的权值。

### 3.3 参数的选取

一般神经网络的权重初始值为随机数,这使得神经网络容易陷入局部极小值,要花大量的时间进行学习调整。而自适应卡尔曼滤波算法的权重值可根据传统卡尔曼滤波的原则确定,在此初值的基础上学习训练,可使网络的收敛速度加快。因此用于估计队列长度的神经元权重初始值选取为  $w_{11} = 1, w_{12} = 1, w_{13}$  为滤波增益矩阵的分量,根据第 3.1 节分析得  $w_{13} = 0.62624$ ;用于估计队列长度变化率的神经元权重初始值选取为  $w_{21} = 1, w_{22}$  为滤波增益矩阵的另一个分量,  $w_{22} = 0.61611$ 。同时通过大量的实验选取动量因子  $\alpha = 0.05$ ,学习

速率  $\eta = 0.5$ 。

## 4 仿真结果及分析

使用 NS-2<sup>[8]</sup> 网络仿真软件验证自适应卡尔曼滤波算法的性能。首先在仿真平台上实现该算法,实验采用如图 4 所示的网络拓扑图进行仿真,并将仿真结果与 PID 算法比较。

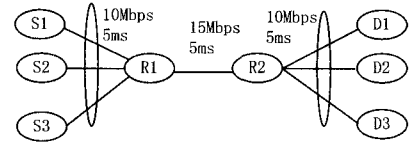


图 4 网络拓扑图

### 4.1 网络拓扑结构与实验参数选择

在此网络拓扑图中, R1 和 R2 是路由器,链路带宽为 15Mbps,延迟为 5ms,平均数据包大小为 500Bytes。除 R1 和 R2 之间的链路以外,其他链路均采用 DropTail 机制,带宽为 10Mb,延迟为 5ms。S1-S<sub>n</sub> 节点连接 N 个持久性的 FTP 流,仿真时间为 60s,各节点的缓冲区大小为 300packets。作为比较,卡尔曼滤波算法的滤波增益矩阵选取为  $K = [0.62426, 0.61611]^T$ 。根据文献[4]选取 PID 的控制参数  $K_p = 2.583911e^{-6}$ ,  $K_i = 2.982912e^{-6}$ ,  $K_d = 3.50735e^{-8}$ ,  $w = 160$ ,  $q_{\text{ref}} = 100$ 。

### 4.2 仿真结果比较及分析

#### 4.2.1 实验 1

实验验证了负载仅为 FTP 响应流时算法的性能。实验开始时,创建 120 个 FTP 流,直到仿真结束,总的仿真时间为 60s。实验结果见图 5 和表 1。PID 算法的队列长度在 [60, 175] 之间波动,平均队列长度为 148.75,队列标准差为 68.47;KF-AQM 算法的队列长度在 [90, 110] 之间,平均队列长度为 102.87,队列标准差为 11.09;adaptive KF-AQM 算法的队列长度在 [90, 105] 之间波动,平均队列长度为 98.31,队列标准差为 5.54。因此 KF-AQM 算法在队列长度稳定性方面性能明显优于 PID 算法,能够快速地将队列长度收敛于期望值附近;但 adaptive KF-AQM 算法又比 KF-AQM 算法的队列波动更小,同时收敛速度较 KF-AQM 也快些。从图中可看出,adaptive KF-AQM 算法的调节时间大约为 1s, KF-AQM 算法的调节时间大约为 2s。

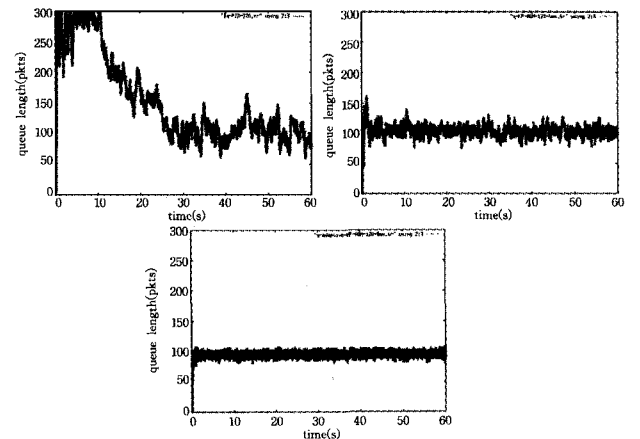


图 5 负载不变时的队列长度变化

表 1 负载不变时的相关性能

|              | PID    | KF-AQM | Adaptive-KF-AQM |
|--------------|--------|--------|-----------------|
| 平均队列长度(pkts) | 148.75 | 102.87 | 98.31           |
| 队列标准差        | 68.47  | 11.09  | 5.54            |

## 4.2.2 实验 2

实验验证了在不同负载变化下算法的性能。总的负载有 270 个 FTP 流,平均分成 3 等份,每隔 20s 启动一组,总的仿真时间为 60s。实验结果见图 6 和表 2。由图 6 可知,PID 算法的队列收敛速度比较慢,而且波动比较大。KF-AQM 算法虽然能在 20s 和 40s 及时地检测出有突发流,但队列长度的收敛性受到影响,40s 后偏离期望的队列长度比较远,平均队列长度为 126.85,同时队列的波动也比在负载不变的情况下更大;而 adaptive KF-AQM 算法在后面 20s 和 40s 有突发流时,检测准确,而且很快地收敛于期望值 100,队列波动也小。从表 2 中看出,adaptive KF-AQM 算法的队列标准差只有 9.77,平均队列长度达到 99.20,说明该算法较其他两种算法最接近期望队列长度,队列稳定性最好。

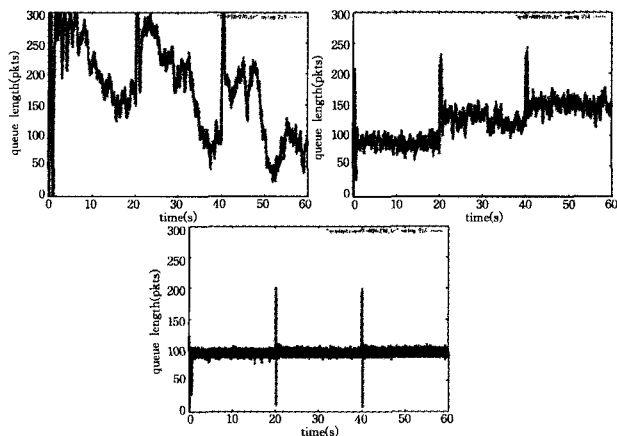


图 6 负载变化时的队列长度变化

表 2 负载变化时的相关性能

|              | PID    | KF-AQM | Adaptive-KF-AQM |
|--------------|--------|--------|-----------------|
| 平均队列长度(pkts) | 171.85 | 126.85 | 99.20           |
| 队列标准差        | 73.29  | 30.97  | 9.77            |

## 4.2.3 实验 3

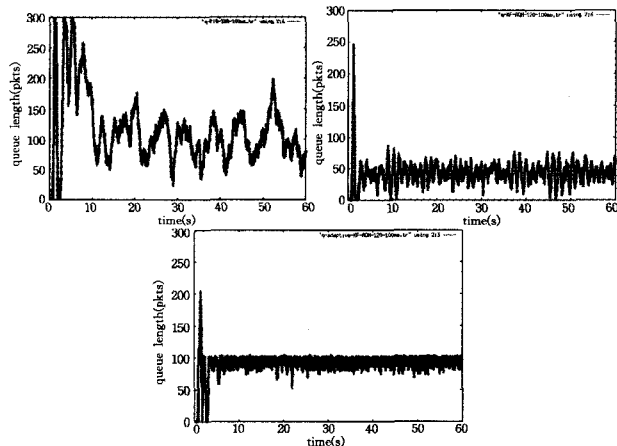


图 7 延时加大时的队列长度变化

本实验验证了算法在大时延网络环境中的性能。实验使用 120 个 FTP 流,直到仿真结束,将 R2 到 D1-Dn 的链路容量加到 45Mbps,延时加大到 100ms,其余仿真环境不变。仿

真结果见图 7 和表 3。可知在这种网络环境下,PID 算法完全不能保持队列的稳定性,队列波动非常大,队列标准差为 58.40;KF-AQM 算法的队列波动虽然没 PID 算法的大,队列标准差也比其更小,但队列长度却远远地偏离了期望值,平均队列长度只有 44.93,远小于 100;而使用 adaptive KF-AQM 算法时,虽然队列长度的抖动比小时延的加大了,收敛速度也比小时延的减慢了,但仍然可以使队列长度维持在期望值附近,平均队列长度有 94.40,较其他两种算法更接近期望值,同时它具有较好的瞬态响应性能,从图中可看出调节时间大约为 3s。

表 3 延时加大时的相关性能

|              | PID    | KF-AQM | Adaptive-KF-AQM |
|--------------|--------|--------|-----------------|
| 平均队列长度(pkts) | 117.79 | 44.93  | 94.40           |
| 队列标准差        | 58.40  | 22.37  | 14.92           |

## 4.2.4 实验 4

模拟真实的网络环境,总的使用了 50 个响应的 FTP 流、10 个非响应的 CBR 流,其中 FTP 流从仿真开始一直到仿真结束。CBR 流分组大小为 1000Bytes,发送速率为 1Mb/s,从 20s 时开启,40s 时结束,持续时间 20s。3 种算法的仿真结果见图 8 和表 4。adaptive KF-AQM 算法稳态性能很好,队列标准差只有 7.86,在 20s 到 40s 时响应速度很快,可很好地反映队列长度的变化情况,基本不受非响应流的干扰;而 PID 算法在 20s 加入非响应流后,完全不能保证队列的稳定性,队列偏差达到 82.60,同时响应流的传输也受到影响,因此该算法在此情况下基本没有可用性;KF-AQM 算法虽然也能在 20s 到 40s 这段时间反映网络的变化情况,但是它的队列长度却偏离期望值较远,平均队列长度只有 85.99。

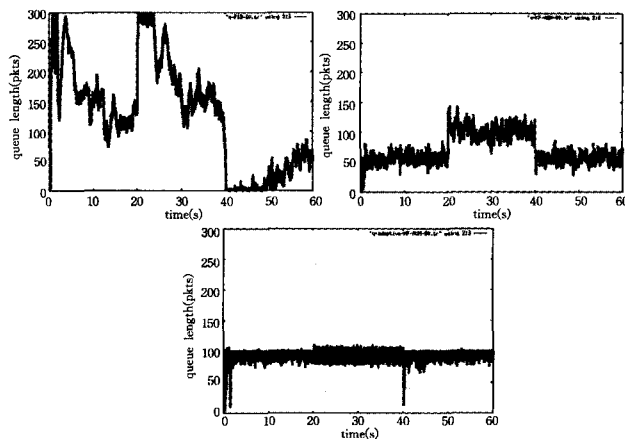


图 8 加入非响应流时的队列长度变化

表 4 加入非响应流时的相关性能

|              | PID    | KF-AQM | Adaptive-KF-AQM |
|--------------|--------|--------|-----------------|
| 平均队列长度(pkts) | 159.18 | 85.99  | 96.11           |
| 队列标准差        | 82.60  | 24.91  | 7.86            |

## 4.2.5 实验结果分析

通过以上实验可知,adaptive KF-AQM 算法是一种基于独立神经网络的智能 AQM 算法,可根据网络传输数据包的多少,自动调节控制器的参数,从而调整队列的丢包率,使队列长度稳定在期望值附近;又因为神经网络的初始连接权重值是依据卡尔曼滤波原则选取的,所以避免了神经网络

(下转第 125 页)

### 5.3 不同网络规模的总体能耗

仿真过程中,选择100~1000个节点,仿真时间设为100s(其它设置与上面实验相同)。图6显示了两种算法在不同网络规模下的总体能耗比率。从图中可以明显看出,当网络规模达到400个节点时,LEACH的总体能耗已经达到了80%,当达到600个节点时网络能量耗尽;而WRMP在200个节点内能耗与LEACH相当,随后能耗一直低于LEACH,到最大1000个节点时还能够剩余20%的能量。这是因为P2P技术的引入,使得网络中的节点能够在分布式的环境下,合理地利用能量资源,更加有效地参与到信息处理的任务中,使WRMP适用于大规模无线传感器网络。

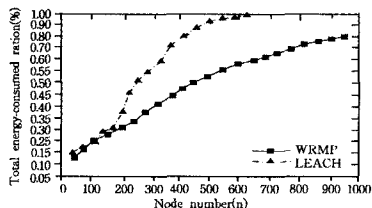


图6 不同网络规模总体能耗的变化情况

**结束语** 本文提出了一种基于WSN架构的完全分布式自组路由模型。通过构建基于网络实际拓扑的P2P网络覆盖层,在LEACH协议上融合结构化CAN算法实现与基站的多跳通信,并在此基础上引入多路径路由技术选择最小能耗路径发送数据,有效提高了大规模WSN路由效率和生命周期。

### 参考文献

- [1] Akyildiz L F, Su W, Sankarasubramanian Y, et al. Wireless sensor networks: a survey[J]. Computer Networks (Elsevier), 2002, 38(4): 393-422
- [2] 唐宏, 谢静, 鲁玉芳, 等. 无线传感器网络原理及应用[M]. 北京: 人民邮电出版社, 2010
- [3] Rondini E, Hailes S, Li L. Load sharing and bandwidth control in

(上接第91页)

陷入局部极小值,加快了学习速率,使算法的收敛速度较快。而PID和KF-AQM算法由于参数固定不变,因此很难适应动态的网络环境,导致队列收敛慢,波动大。

**结束语** 主动队列管理是近几年端到端拥塞控制中研究的一个热点。提出了一种新的自适应主动队列管理算法,它将现代控制理论中的卡尔曼滤波原理与神经元网络结合,在线调整滤波增益矩阵各个分量,使队列稳定在期望值附近。仿真结果表明,在大多数网络环境中,本算法在队列稳定性、延时和链路利用率等方面,性能明显优于PID和KF-AQM算法,同时可以保证快速的收敛速度和较好的瞬态性能。因此,自适应卡尔曼滤波算法是一种有效的主动队列管理算法。

### 参考文献

- [1] Jacobson F S. Random early detection gateways for congestion avoidance[J]. IEEE/ACM Transactions on Networking, 1993, 1(4): 397-413
- [2] Hollot C V, Misra V. On Designing Improved Controllers for

mobile P2P wireless sensor networks[C]//Sixth Annual IEEE International Conference on Pervasive Computing and Communications. IEEE, 2008: 468-473

- [4] Chellury S, Chi M, Michael L, et al. Peer-to-peer Wireless Sensor Network Data Acquisition System with Pipelined Time Division Scheduling[C]//Sarnoff Symposium, IEEE, 2006. Princeton, NJ, USA: IEEE, 2006: 1-4
- [5] LI Li, Liu Yuan-an, Tang Bi-hua. SNMS: an intelligent transportation system network architecture based on WSN and P2P network[J]. The Journal of China Universities of Posts and Telecommunications, 2007, 14(1): 65-70
- [6] 张丽娟. 结构化无线传感器网络路由协议的研究[D]. 大连: 大连理工大学, 2009
- [7] Gutierrez G, Mejias B, van Roy P, et al. WSN and P2P: A Self-Managing Marriage[C]//2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops. Venice, Italy: SASO, 2008: 198-201
- [8] Xue W, Sheng W, Dao W B, et al. Distributed peer-to-peer target tracking in wireless sensor networks[J]. Sensors, 2007, 7(6): 1001-1027
- [9] Carretti C M. Comparison of distributed optimization algorithms in sensor networks[M]. Stockholm, Sweden: Royal Institute of Technology(KTH), 2008
- [10] 崔瑞端, 庄雷. 基于P2P的小型无线传感器网络的研究[J]. 传感技术学报, 2006, 19(3): 900-904
- [11] Heinzelman W, Chandrakasan A. Energy-Efficient Communication Protocol for Wireless Microsensor Networks[C]//Proceedings of the 33rd Hawaii International Conference on System Sciences. Washington: IEEE Press, 2000: 3005-3014
- [12] Ratnasamy S, Francis P, Handley M, et al. A Scalable Content-Addressable Network[C]//Proceedings of the International Conference of the Special Interest Group on Data Communication(SIGCOMM '01). New York, NY, USA: ACM Press, 2001: 161-172

AQM Routers Supporting TCP Flows[C]//Proc of INFOCOM'01. Anchorage, USA, 2001: 1726-1734

- [3] Fan Yan-fei, Ren Feng-yuan, Lin Chuang. Design a PID Controller for Active Queue Management[C]//Proc of 8th IEEE International Symposium on Computers and Communication. Antalya, Turkey, 2003: 985-990
- [4] Yan Qiao, Hu Xiao-juan. A New PID Controller for AQM Based on Neural Network[C]//Proceeding of Intelligent Computing and Intelligent Systems(ICIS). 2010 IEEE International, 2010: 804-808
- [5] Alinaghizadeh M, Beheshti M T H. A Robust Discrete-time Controller for Delay Sensitive Applications[J]. IEICE Electronics Express, 2009, 22(6): 1548-1555
- [6] Cotter S F, Murthi M N. Target tracking based network active queue management[C]//Proc of IEEE International Conference on Acoustics, Speech and Signal Processing. Taipei, 2009: 2757-2760
- [7] 刘金琨. 智能控制[M]. 北京: 电子工业出版社, 2005
- [8] USC/ISI, Los Angeles, CA. The NS simulator and the documentation[EB/OL]. <http://www.isi.edu/nsnam/ns/>