

文章编号:1673-9469(2011)02-0100-05

## 基于 NLMS 算法的自适应 AQM 控制机制的研究

郜亚丽<sup>1</sup>,李世勇<sup>2</sup>

(1. 济源职业技术学院 实验实训中心,河南 济源 454650;2. 北京交通大学 电子信息工程学院 下一代互联网互联设备国家工程实验室,北京 100044)

**摘要:**针对 RED 算法在网络拥塞控制中使得队列波动较大的缺点,提出了一种归一化最小均方(NLMS)算法,并研究了 NLMS 算法在主动队列管理中的具体实现。仿真实验通过对 NLMS 算法、RED 算法、REM 算法、LRC-RED 算法性能的比较,表明 NLMS 算法具有较好的动静态性能,且能够提高队列稳定性,降低丢包率。

**关键词:**控制机制;归一化最小均方算法;主动队列管理;队列稳定性

**中图分类号:** TP393

**文献标识码:** A

## Study of NLMS algorithm based AQM control mechanism

GAO Ya-li<sup>1</sup>, LI Shi-yong<sup>2</sup>

(1. Center of Training & Experimentation, Jiyuan Vocational and Technical College, He'nan Jiyuan 454650, China;  
2. National Engineering Laboratory of Next Generation Internet Interconnection Devices, School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China)

**Abstract:** RED algorithm for network congestion control causes the shortcomings of the queue fluctuations, a normalized least mean square (NLMS) algorithm was proposed, and the NLMS algorithm in active queue management in the implementation was introduced. The performances of NLMS algorithm, RED algorithm, REM algorithm, and LRC-RED algorithm were compared by simulation experiments. The results show that the NLMS algorithm has good dynamic and static performance, and it can improve the stability of the queue and reduce the packet loss rate.

**Key words:** congestion control; NLMS algorithm; active queue management; queue stability

随着 Internet 网络规模的迅速扩大,网络上开放的业务种类不断增加,网络应用的不断深入,导致网络吞吐量急剧降低,严重时甚至发生网络崩溃,这就是网络拥塞现象。网络拥塞已经成为制约网络发展和应用的瓶颈。随着网络规模的增大,仅仅依靠 TCP 拥塞控制机制来提高网络的服务质量是远远不够的,因此路由器作为网络的中间节点也必须参与到网络拥塞控制中来。近年来,主动队列管理(active queue management,简称 AQM)<sup>[1]</sup>成为网络拥塞控制研究中的一个技术热点。它通过网络中间节点由控制的分组丢弃机制,实现了较低的排队延时和较高的有效吞吐量。研究人员提出了多种 AQM 算法,如 RED<sup>[2]</sup>, REM<sup>[3]</sup>, PI<sup>[4]</sup>, LRC-RED<sup>[5]</sup>等。最早经典的当数

由 Floyd 等于 1993 年提出的随机早期丢弃 RED (Random Early Drop)算法。该算法是目前最常用的一种 AQM 算法。

RED 算法以平均队列长度作为拥塞指示来控制包的丢失。在动态网络中,这些算法对突发流不敏感,使得队列长度波动较大<sup>[6]</sup>。本文通过引入加权队列长度作为拥塞指示,使用归一化最小均方(NLMS)算法,结合对分组丢弃概率的更合理的计算,将瞬时队列长度控制在一个较为稳定的范围内。并通过算法仿真实验和性能比较,验证了该算法在保持队列稳定的同时丢包率也有所降低。

### 1 归一化最小均方(NLMS)算法

LMS (Least mean square)中文是最小均方算

收稿日期:2011-02-20

基金项目:国家自然科学基金(60903150)

作者简介:郜亚丽(1975-),女,河南济源人,工程硕士,讲师,从事计算机网络应用与多媒体技术。

法,是经典的自适应滤波器算法<sup>[7]</sup>,具有实现简单,计算量小等优点。原理图如图 1 所示,分为波束赋形和自适应权重控制两个部分,通过迭代的方法来求解 MMSE 准则下的最优权重。NLMS 算法是改进的 LMS 算法,又称为归一化最小均方算法,是采用变步长的方法来缩短自适应收敛过程。

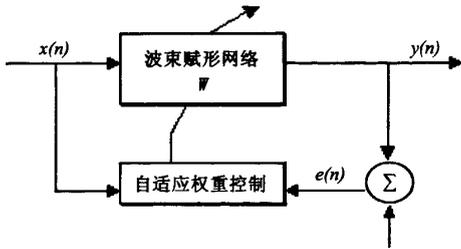


图1 自适应滤波器原理图

Fig.1 Principle diagram of adaptive filter

以自适应系统辨识<sup>[8]</sup>为例,  $x(n)$  是输入参考信号,  $y(n)$  是未知系统的输出信号。则自适应滤波器输出的  $y(n)$  的预估值为

$$\hat{y}(n) = \sum_{i=0}^{N-1} h_i(n)x(n-i)$$

式中  $N$ —滤波器的阶数;  $\{h_i(n) | i = 0, 1, \dots, N-1\}$ —第  $n$  次迭代后自适应滤波器的系数。

系统估值误差  $e(n) = y(n) - \hat{y}(n)$ , 滤波器参数的更新公式为

$$h_i(n+1) = h_i(n) - \frac{\mu \partial e^2(n)}{2 \partial h_i(n)} = h_i(n) + \frac{\mu e(n)x(n-i)}{a + \sum_{j=0}^{N-1} x^2(n-j)} \quad (1)$$

式中  $\mu$ —步长,为了保证算法收敛,应满足  $0 < \mu \leq 2$ ;  $a$ —一个小的正数,它保证分母不为零;  $\frac{\partial e^2(n)}{\partial h_i(n)} = -2e(n)x(n-i)$ —系数  $h_i(n)$  的梯度。

该算法在输入信号较大的情况下避免梯度噪声放大的干扰,因而具有较好的收敛性能<sup>[8]</sup>。

## 2 NLMS 算法在主动队列管理中的实现

### 2.1 算法的描述

基于 NLMS 算法的基本原理是:引入加权队列长度作为拥塞指示,通过对过去  $N$  个时刻的瞬时队列长度信息分别赋以不同权重,并采用 NLMS 算法中的参数调整方法对权重自适应调整来得到加权队列长度。由于权重因子可以自适应调节,且

采用了过去时刻的瞬时队列值,因此加权队列长度比 RED 算法中的平均队列长度更及时地反映网络流量变化情况,从而对拥塞作出反应。

算法的实现主要分三步:

第一步:计算加权队列长度。第二步:结合加权队列长度和网络负载流量进行丢包决策。第三步:采用 NLMS 的方法更新权值,回到第一步。

### 2.2 加权队列长度计算及权值更新

$$\text{加权队列长度 } wq = \sum_{i=0}^{N-1} w(n-i)q(n-i)$$

式中  $wq$ —加权队列长度;  $w(n-i)$ —过去第  $i$  个时刻的瞬时队列值所占的权重;  $q(n-i)$ —过去第  $i$  个时刻的瞬时队列值。

权重因子在加权队列长度和当前瞬时队列长度之差的基础上动态更新。误差  $e(n)$  计算为  $e(n) = q(n) - wq$ , 由式(1)得到队列权重的更新

$$\bar{w}(n+1) = \bar{w}(n) + \mu \bar{q}(n)e(n) \quad (2)$$

$\mu$  为 NLMS 的比例因子,由下式决定

$$\mu = \frac{\mu_0}{a + q^r(n)q(n)} \quad (3)$$

式中  $\mu_0$  取 1 得以保证收敛,  $a$  取 1 得以保证分母不为零。

### 2.3 丢包决策

结合加权队列长度和网络负载流量进行丢包决策。由于链路负载信息能在一定程度上反映拥塞,采用加权队列长度和链路负载信息作为拥塞指示,以此管理丢包率。负载信息使用“负载因子”来衡量。所谓负载因子就是在网络中,某条链路上的包到达速率与该链路的带宽之比  $\lambda = \frac{\gamma}{C}$ 。

式中  $\lambda$ —负载因子(load factor);  $\gamma$ —包到达速率;  $C$ —链路的服务速率(即链路带宽)。

在进行分组丢弃概率计算时,考虑了链路负载和队列长度信息,分组的丢弃概率  $p$  计算如下

$$p = \left( \frac{\gamma - C}{\gamma} \right) \frac{wq}{B} = \left( 1 - \frac{1}{\lambda} \right) \frac{wq}{B} \quad (4)$$

式中  $wq$ —估计的队列长度;  $B$ —缓存的大小。

通过这种概率丢弃使得(被接纳的)包到达速率与链路容量达到平衡,同时还考虑了队列长度,队列长度越长则分组被丢弃的概率也越大。

具体丢包策略如下:

(1)当  $\lambda < 1$  且  $wq < \frac{1}{2}B$  时,网络处于轻负载区,所有分组丢弃概率为 0。

(2)当  $\lambda < 1$  且  $wq > \frac{1}{2}B$  时,网络处于轻负载区,此时小概率丢包,  $P = P_{\max} \frac{wq}{B}$ 。

(3)当  $\lambda > 1$  且  $wq > \frac{1}{2}B$  时,网络处于或者即将处于拥塞区,需及时采取控制,对到达的分组按照式(4)丢包。

### 3 仿真实验与性能比较

利用网络仿真软件 NS-2 来验证算法的性能。采用 NLMS(实验中称为 NAQM)算法、RED 算法、REM 算法、LRC-RED 算法进行仿真比较。实验环境为多瓶颈链路,实验网络拓扑结构如图 2 所示。在图 2 中,从左到右的五条链路带宽均为 15M,延时 20ms。sender 为发送端,receiver 为接收端。其中, sender1 到 receiver1 是 100 个 TCP 流, sender2 到 receiver2 是 30 个 TCP 流, sender3 到 receiver3 是 30 个 TCP 流。在实验中,瓶颈链路 r2-r3 的特性和 r4-r5 的特性类似,而链路 r1-r2、r3-r4、r5-r6 基本不会出现拥塞。因此仅分析 r2-r3 之间的性能。

#### 3.1 负载固定情况下的性能

实验中,考虑随时间变化负载固定的情况下,

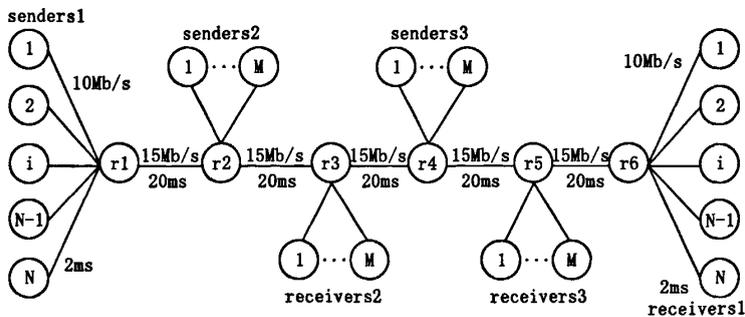


图2 仿真网络拓扑图

Fig.2 Simulation network topology

表 1 静态时各算法在链路 r2-r3 之间的丢包率

Tab.1 The packet loss rate of each algorithm between link r2 and link r3 in static state

AQM 算法	发送的包总数	丢弃的包总数	平均丢包率
RED	93 364	3 132	3.354%
REM	94 410	4 020	4.258%
LRC-RED	93 657	3 551	3.791%
NAQM	93 390	3 159	3.382%

考察各算法在队列长度的稳定性、丢包率大小的变化。其中, senders1, senders2, senders3 分别同时启动 100 个, 30 个, 30 个 TCP 连接, 仿真时间为 50s。下面是各种性能指标的仿真结果。

队列长度的变化。图 3 所示的是在负载固定的情况下,各算法在维持队列稳定性方面的性能。由图中四种算法的对比可以看出,在多瓶颈链路中,NAQM 算法能够很好的维持在 200 附近,并且波动较小,因此稳定性也最好。LRC-RED 算法的队列长度也基本保持在 200 附近,但队列的波动比 NAQM 算法稍大一些。RED 算法所维持的队列长度波动较大。REM 算法基本处于满队列,无法保持在期望值附近,且波动大。由此可见,在多瓶颈链路中,负载固定时,NAQM 算法所表现出的性能是最好的,其队列最稳定,能够很好的保持在期望值附近,队列波动小,且响应时间短。

丢包率的变化。表 1 所示为静态情况下节点 r2-r3 之间的链路的各算法丢包率。通过表中四个算法的比较,RED 与 NAQM 算法的丢包率较为接近,约为 3.3% 左右。其次是 LRC-RED,丢包率最大的是 REM 算法。

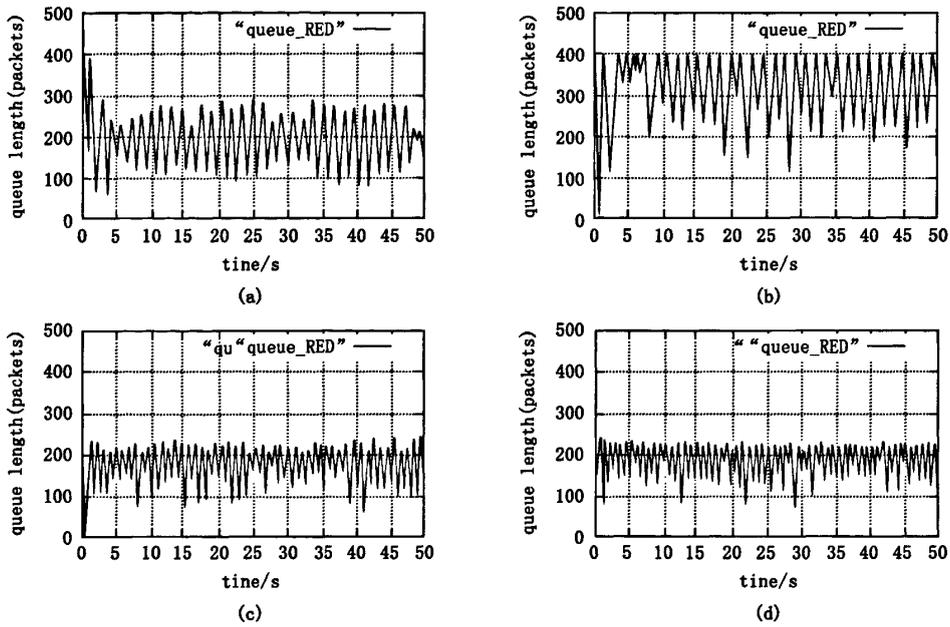


图 3 负载不变的情况下各算法的队列长度变化

Fig. 3 The queue length change of each algorithm under invariable load condition

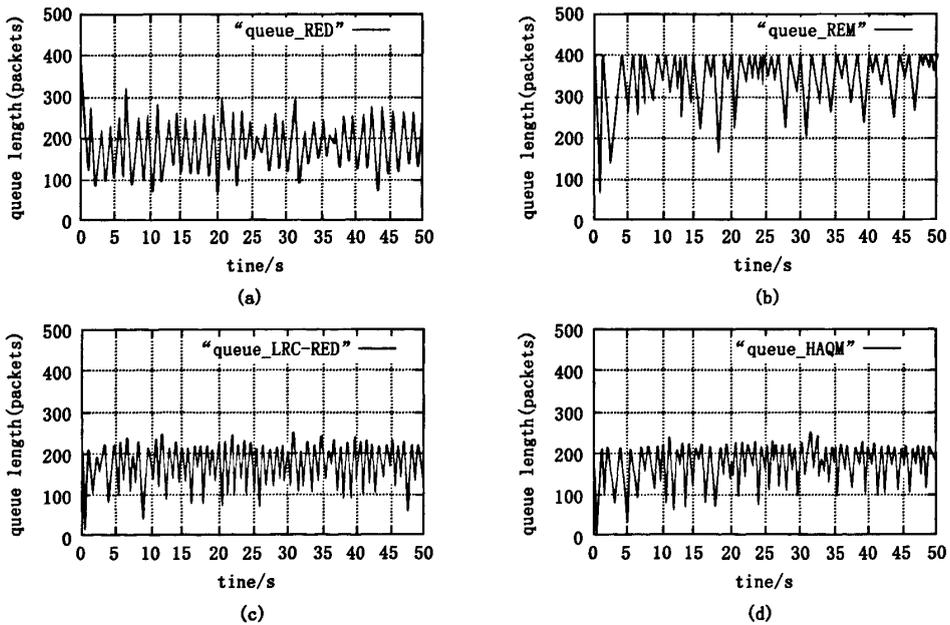


图 4 负载变化的情况下各算法的队列长度变化

Fig. 4 The queue length change of each algorithm under variable load condition

### 3.2 负载变化环境下的性能

实验中,考虑负载随时间变化的情况下,考察各算法在队列长度的稳定性、丢包率大小的变化

方面的性能。其中,senders1 分别在 0s,5s,10s,20s,30s 启动 20 组 TCP 连接,senders2、senders3 分别在 0s 启动 30 组 TCP 连接,仿真时间为 50s。下面是各种性能指标的仿真结果。

表 2 动态时各算法在链路 r2 - r3 之间的丢包率

Tab.2 The packet loss rate of each algorithm between link r2 and link r3 in dynamic state

算法	发送的包总数	丢弃的包总数	平均丢包率
RED	92 824	2 446	2.635%
REM	93 196	2 688	2.884%
LRC - RED	93 317	3 015	3.231%
NAQM	93 004	2 694	2.896%

队列长度的变化。图 4 所示的是在负载变化的情况下,各算法在维持队列稳定性方面的性能。由图中四种算法的对比可以看出,在动态多瓶颈链路中,NAQM 算法能够很好地维持在 200 附近,并且波动较小。LRC - RED 算法次之,RED 算法的队列长度波动较大,并且在每次增加负载时,队列有波动。REM 算法基本处于满队列,无法保持在期望值附近。由此可见,在多瓶颈链路中,负载变化时,NAQM 算法所表现出的性能是最好的,其队列最稳定,能够很好地保持在期望值附近,队列波动小,且响应时间较短。

丢包率的变化。表 2 所示为动态情况下节点 r2 - r3 之间链路的各算法丢包率。通过表中四个算法的比较,RED 算法的丢包率最小,REM 和 NAQM 算法的丢包率相近,约为 2.8% 左右。丢包率最大的是 LRC - RED 算法,约为 3.2% 左右。

#### 4 结论

1) 主动队列管理在保证高吞吐量的同时,能有效控制缓冲队列的长度,减小网络时延。

2) 采用归一化最小均方(Normal Least Mean Square, NLMS)的方法对权值自适应调整,结合负载因子对分组进行更为合理的丢弃,将队列长度

的变化稳定在一个理想的水平。

3) 仿真实验表明该算法具有较好的动静态性能,且能提高队列稳定性,降低丢包率。尤其在多瓶颈链路中,算法的队列稳定性最好。

#### 参考文献:

- [1] 朱小艳,李向丽.主动式队列管理(AQM)算法研究[J].微计算机信息,2006(1):2-3.
- [2] 魏涛,张顺颐.一种模糊自调整的 PD-RED 算法[J].计算机工程与应用,2007,43(5):124-126.
- [3] 苏聪,陈元琰,罗晓曙.基于模糊理论的主动队列管理算法-FBLUE[J].计算机工程与应用,2006,42(23):117-120.
- [4] 朱华,向少华.一种模糊自适应 PI 算法在网络拥塞控制中的应用[J].大众科技,2009,123(11):32-34.
- [5] 任丰原,林闯,刘卫东.IP 网络中的拥塞控制[J].计算机学报,2003,26(9):1025-1034.
- [6] 薛质,潘理,李建华.基于模糊 RED 算法的 IP 拥塞控制机制[J].计算机工程,2002,28(3):60-61.
- [7] 谷源涛,唐昆,崔慧娟,等.变步长归一化最小均方算法[J].清华大学学报(自然科学版),2002,42(1):15-18.
- [8] HAYKIN S.自适应滤波器原理(第三版)[M].北京:电子工业出版社,1998.

(责任编辑 刘存英)