

基于动态矩阵控制的主动队列管理算法

徐 琴, 孙金生

(南京理工大学自动化学院, 江苏南京 210094)

摘 要: 针对 Internet 系统, 通过对流体流模型的分析, 提出了一个新的预测模型. 该模型形式简单, 参数的计算相对容易, 并且能根据当前的网络情况有效的预测拥塞窗口的变化. 结合动态矩阵控制 (Dynamic Matrix Control, DMC) 理论, 提出了一种新的主动队列管理算法——DMCAQM 算法, 给出了 DMCAQM 的详细设计过程, 稳定性分析和参数选取原则. 大量不同网络环境的仿真实验表明 DMCAQM 算法是有效的. 与 PI, RaQ 和 REM 等算法相比较, DMCAQM 有收敛速度快、队列抖动小的优点. 同时, 由于 DMCAQM 的采样间隔相对较大, 而算法实现简单, 所以计算量小, 占用的路由资源更少.

关键词: 网络拥塞控制; 主动队列管理算法; 动态矩阵控制

中图分类号: TP393

文献标识码: A

文章编号: 0372-2112 (2014)05-0833-08

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2014.05.001

A New Active Queue Management Algorithm Based on Dynamic Matrix Control

XU Qin, SUN Jin-sheng

(School of Automation, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China)

Abstract: Through the analysis of the fluid-based model, a new predictive model is proposed for Internet system. This model is simple, and the way of selecting parameters is relatively easy. It's able to predict the change of congestion window according to the network environment. Based on the theory of Dynamic Matrix Control (DMC), a new Active Queue Management (AQM) algorithm is proposed, called DMCAQM. The design details, the stability analysis and the principles of choosing parameters are also presented. The results of simulations under different network environments demonstrate that DMCAQM is effective in controlling the queue length to the desired value. In comparison to other AQM algorithms, such as PI, RaQ and REM, DMCAQM have faster response and lower queue fluctuations. In addition, DMCAQM has larger interval while with simple implementation, so it use less resource on the routers.

Key words: network congestion control; active queue management; dynamic matrix control

1 引言

随着网络的爆炸式增长, 网络拥塞问题日趋严峻. 为了更好地解决这个问题, 人们越来越多的关注网络拥塞控制的研究. 其中主动队列管理算法 (Active Queue Management, AQM) 作为网络拥塞控制重要部分, 得到了良好的发展. 最初, 研究者们提出的算法简单直观, 如 RED^[1], AVQ^[2] 等. 这类算法实现简单, 但是由于缺乏理论依据, 存在着很多问题, 如固定的参数难以适应众多不同网络环境等. 随着基于流体流模型^[3]的提出, 控制理论被引入 AQM 算法的分析, 为 AQM 算法的研究打开新的篇章^[4,5].

虽然控制算法种类繁多, 却并不都适合 AQM 算法的设计. 这是由于对于中间节点而言网络负载、传输时延等是未知的, 这意味着网络模型中的部分参数未知. 实际上网络环境多变, 这使得网络环境参数更加难以确定. 而预测控制具有良好的鲁棒性, 对存在延时的系统控制效果也较好, 因此对于网络拥塞控制而言, 预测控制是理想的选择.

近几年预测控制已在 AQM 算法的研究中得到了应用. Bigdeli 等将线性化后的流体流模型化成离散模型. 然后结合预测函数控制的思想, 提出了 PFC-AQM 算法^[6]. 然而其模型参数的计算方式复杂, 不易于实现. Marami 等利用模型预测控制 (Model Predict Controller,

MPC) 分析网络预测未来的输出并计算最优控制律^[7], 提出了新的 AQM 算法. 但其实现复杂, 甚至需要现场可编程门阵列(Field-Programmable Gate Array, FPGA) 的辅助, 如果要用于实际网络环境, 需要改进其实现方式. Wang 等提出了 DM-AQM 算法^[8], 该算法需要寻找稳态工作点才能获得网络模型的线性表达式, 这意味着不同的网络环境需要多次寻找稳态工作点然后获得模型, 而在网络环境随着时间变化, 稳态工作点也是变化的, 实时的寻找稳态工作点是很难实现的. 为了克服了对稳态工作点的依赖, 他们将流体流模型线性化后看作反馈控制模型, 利用模型预测(Model Predict, MP) 控制思想, 设计状态反馈观测器和反馈控制器, 提出了 MPAQM 算法^[9]. 然而其模型参数计算复杂, 状态量包含未来的输出, 需要设计状态观测器, 使算法实现复杂的同时也引入了干扰.

可以看出预测控制虽然理论上适合网络拥塞控制的研究, 然而网络拥塞控制模型的局限性成为这方面研究的难点. 主要表现为两个方面, 模型参数计算复杂和实现需要的资源庞大. 为了更好的解决网络拥塞问题, 本文在流体流模型的基础上, 通过研究分析, 提出了新的预测模型, 该模型避免了需要预测的状态变量, 同时参数的选择相对简单. 利用动态矩阵控制(Dynamic Matrix Control, DMC) 的思想, 求出最优控制律, 提出了一种新的 AQM 算法——DMCAQM. 该算法实现简单, 占用的资源较少, 并且能够快速有效的控制队列长度, 使其稳定在期望值附近.

2 DMCAQM 算法

2.1 预测模型

Misra 等^[3]借助动力学的相关理论提出了著名的流体流模型:

$$\begin{cases} \dot{W}(t) = \frac{1}{R(t)} - \frac{W(t-R)W(t)}{2R(t)}p(t-R) \\ \dot{q}(t) = -C + N \frac{W(t)}{R} \end{cases} \quad (1)$$

其中 W 是平均 TCP 拥塞窗口的大小, C 是链路容量, N 是 TCP 连接的个数, 即网络负载, q 是缓冲队列长度, p 是数据包丢弃概率或者标记概率(采用包标记), R 是往返时延, 它由排队时延和传输时延决定, 即 $R(t) = R_{pt} + q(t)/C$. 其中 R_{pt} 代表往返传输延时.

Hollot 等^[10]计算平衡点(W_0, q_0, p_0)如下:

$$\begin{cases} \dot{W} = 0 \Rightarrow W_0^2 p_0 = 2 \\ \dot{q} = 0 \Rightarrow W_0 = \frac{R_0 C}{N} \end{cases} \quad (2)$$

并将流体流模型在平衡点(W_0, q_0, p_0)附近线性化, 有

$$\begin{aligned} \delta \dot{W}(t) = & -\frac{N}{R_0^2 C}(\delta W(t) + \delta W(t-R_0)) \\ & -\frac{R_0 C^2}{2N^2} \delta p(t-R_0) \end{aligned} \quad (3)$$

其中 $\delta W = W - W_0$, $\delta p = p - p_0$, $R_0 = R_{pt} + q_{ref}/C$, q_{ref} 代表期望队列长度.

为了方便后面的说明, 将 δW 改写为 y , δp 改写为 u , 则有:

$$\frac{dy(t)}{dt} = -\frac{N}{R_0^2 C}(y(t) + y(t-R_0)) - \frac{R_0 C^2}{2N^2} u(t-R_0) \quad (4)$$

其中 dy 和 dt 分别代表 y 和 t 的微分. 当 dt 足够小时, $dy(t) = y(t+dt) - y(t)$, 则:

$$\begin{aligned} y(t+dt) = & y(t) - \frac{N}{R_0^2 C}(y(t) + y(t-R_0))dt \\ & - \frac{R_0 C^2}{2N^2} u(t-R_0)dt \end{aligned} \quad (5)$$

假设采样间隔为 dt , 令 $m = \lceil \frac{R_0}{dt} \rceil$, 即 m 取大于 $\frac{R_0}{dt}$ 的最接近的整数. 将上述连续模型写成离散的形式, 则:

$$y(k+1) = a_1 y(k) + a_2 y(k-m) + a_3 u(k-m) \quad (6)$$

其中 $a_1 = 1 - \frac{N}{mR_0 C}$, $a_2 = -\frac{N}{mR_0 C}$, $a_3 = -\frac{R_0 C^2}{2N^2 m}$. 那么第 i 个采样时刻的预测值为:

$$\begin{aligned} y(k+i|k) = & a_1 y(k+i-1) + a_2 y(k-m+i-1) \\ & + a_3 u(k-m+i-1) \end{aligned} \quad (7)$$

整理上述等式, 则有:

$$\begin{aligned} (1 - a_1 z^{-1} - a_2 z^{-m-1}) \hat{y}(k+i) \\ = a_3 z^{-m-1} u(k+i) \end{aligned} \quad (8)$$

其中 $\hat{y}(k+i)$ 代表第 k 个采样时刻对 $y(k+i)$ 的估计值, 即 $y(k+i|k)$. 由上式可以看出, 为了估计 $\hat{y}(k+i)$, 需要估计 $\hat{y}(k+1), \hat{y}(k+2), \dots, \hat{y}(k+i-1)$ 等多个值. 为了避免需要预测的状态变量, Clarke 等^[11]通过引入一个特殊的 Diophantine 方程将需要预测的状态量变换成已知的输入和输出量, 解决了这种预测控制中的问题. 本文借鉴这种方式, 引入一个特殊的 Diophantine 方程如下:

$$(1 - a_1 z^{-1} - a_2 z^{-m-1}) E_i(z^{-1}) = 1 - z^{-i} F_i(z^{-1}) \quad (9)$$

其中, $E_i(z^{-1})$ 是阶次不大于 $i-1$ 的多项式, $F_i(z^{-1})$ 是阶次为 m 的多项式.

式(8)两边同时乘以 $E_i(z^{-1})$, 稍作整理则有:

$$\hat{y}(k+i) = z^{-i} F_i(z^{-1}) \hat{y}(k+i) + a_3 E_i(z^{-1}) z^{-m-1} u(k+i) \quad (10)$$

其中, 等式右边第一部分是过去的可测量的输出, 当 $i > m$ 时, 第二部分包含过去和将来的输入, 令:

$$\hat{y}(k+i) = \Gamma_i(k) + \mathbf{B}_i \mathbf{U}(k) \quad (11)$$

其中 Γ_i 代表过去的输入与输出, $\mathbf{B}_i \mathbf{U}(k)$ 代表将来的输

入.其中 $\mathbf{B}_i = [b_1, b_2, \dots, b_n]$,

$\mathbf{U}(k) = [u(k), u(k+1), \dots, u(K+n-1)]^T$, n 为控制时域.

按照上述方式,求出 $i = m+1, m+2, \dots, m+n$ 时式(11)的各项系数,可以获得一系列等式,将其整理为如下矩阵形式:

$$\hat{\mathbf{Y}}(k) = \mathbf{\Gamma}(k) + \mathbf{B}\mathbf{U}(k) \quad (12)$$

其中

$$\hat{\mathbf{Y}}(k) = [\hat{y}(k+m+1), \hat{y}(k+m+2), \dots, \hat{y}(k+m+n)]^T,$$

$$\mathbf{\Gamma}(k) = [\Gamma_{m+1}(k), \Gamma_{m+2}(k), \dots, \Gamma_{m+n}(k)]^T,$$

$$\mathbf{B} = [B_{m+1}, B_{m+2}, \dots, B_{m+n}]^T.$$

式(12)是后面要用到的预测模型.其中 $\hat{\mathbf{Y}}(k)$ 预测的将来的窗口的变化,而 $\mathbf{U}(k)$ 则是丢弃(或标记)概率的变化.对于路由器来说,窗口不能直接测量,然而其数据包到达率直观上是与 TCP 连接的数据包发送率相等,所以目前比较认可的方式是通过数据包到达率估计其值,即 $W(k) = \frac{g(k)R_0}{Ndt}$.其中 $g(k)$ 是第 k 个采样间隔中数据包到达的数量,在中间节点上可以直接测量获得.

由于 $y(k)$ 代表 δW ,可以得到 $y(k) = (\frac{g(k)}{dt} - C)R_0/N$.

2.2 滚动优化

AQM 算法最重要的控制目标是路由器的缓冲队列稳定,这意味着稳态时数据包到达率波动越小越好,丢弃概率的波动也不宜过大.所以,DMCAQM 的性能指标如下:

$$\min J(k) = \|\mathbf{\Omega}(k) - \hat{\mathbf{Y}}(k)\|_{\mathbf{Q}}^2 + \|\mathbf{U}(k)\|_{\mathbf{S}}^2 \quad (13)$$

其中 $\mathbf{\Omega}(k) = [\omega(k+m+1), \omega(k+m+2), \dots, \omega(k+m+n)]^T$ 是 $\hat{\mathbf{Y}}(k)$ 的期望值,其具体形式会在下一小节给出, $\mathbf{Q} = \text{diag}(q_1, q_2, \dots, q_n)$ 是误差加权矩阵, $\mathbf{S} = \text{diag}(s_1, s_2, \dots, s_n)$ 是控制作用加权矩阵.

将式(12)带入式(13),求得:

$$\mathbf{U}(k) = (\mathbf{U}^T \mathbf{Q} \mathbf{B} + \mathbf{S})^{-1} \mathbf{B}^T \mathbf{Q} [\mathbf{\Omega}(k) - \mathbf{\Gamma}(k)] \quad (14)$$

上式给出了 $u(k), u(k+1), \dots, u(k+n-1)$ 的最优值,但是实现时只取 $u(k)$,每次采样都重新计算 $u(k)$.

2.3 校正

在预测模型式(12)中,没有直接涉及队列长度,所以容易产生稳态误差.为了避免稳态误差,将模型修正与参考轨迹相结合,获得新的参考轨迹的形式如下:

$$\mathbf{\Omega}(k) = (q_{ref} - q(k) - \beta(k)) \frac{R_0}{Nm} \mathbf{H} \quad (15)$$

其中 q_{ref} 为参考队列长度, $\mathbf{H} = [h_1, h_2, \dots, h_n]^T$ 是校正向量, $\beta(k)$ 有如下形式:

$$\beta(k) = \beta(k-1) + \mu(q(k) - q_{ref}) \quad (16)$$

其中 μ 为校正系数.

3 稳定性分析及参数选择

本节将用控制原理来分析采用 DMCAQM 算法的

TCP/AQM 系统的稳定性,并给出系统稳定的充要条件.同时根据系统稳定的条件,给出参数设置的原则和方法.

3.1 稳定性分析

为了方便对 TCP/AQM 系统进行分析,首先将 $y(k)$ 转化为与 $q(k)$ 有关的形式如下:

$$\begin{aligned} y(k) &= (\frac{g(k)}{dt} - C)R_0/N \\ &= (q(k+1) - q(k)) \frac{R_0}{Nm} \end{aligned} \quad (17)$$

令 $y_r(k) = (q_{ref} - q(k)) \frac{R_0}{Nm}$,根据式(15)有:

$$\begin{aligned} \mathbf{\Omega}(k) &= (y_r(k) + \mu \sum_{i=1}^k y_r(i)) \frac{R_0}{Nm} \mathbf{H} \\ &= F(z^{-1}) y_r(k) \mathbf{H} \end{aligned} \quad (18)$$

其中 $F(z^{-1}) = (1 + \mu + \mu z^{-1} + \dots + \mu z^{-k})$.由于 $\mathbf{\Gamma}(k)$ 过去的输入输出,将其改写为如下形式:

$$\mathbf{\Gamma}(k) = \mathbf{\Gamma}_y(z^{-1}) y(k) + \mathbf{\Gamma}_u(z^{-1}) u(k) \quad (19)$$

令 \mathbf{L}^T 为 $(\mathbf{B}^T \mathbf{Q} \mathbf{B} + \mathbf{S})^{-1} \mathbf{B}^T \mathbf{Q}$ 的第 1 行,根据式(14)有:

$$\begin{aligned} u(k) &= \mathbf{L}^T (F(z^{-1}) y_r(k) \mathbf{H} - \mathbf{\Gamma}_y(z^{-1}) y(k) \\ &\quad - \mathbf{\Gamma}_u(z^{-1}) u(k)) \end{aligned} \quad (20)$$

定理 1 由式(6)表示的系统稳定当且仅当 $(1 + \mathbf{L}^T \mathbf{\Gamma}_u(z^{-1})) (1 - a_1 z^{-1} - a_2 z^{-m-1}) + a_3 z^{-m-1} \mathbf{L}^T \mathbf{\Gamma}_y(z^{-1})$ 的所有解在单位圆内.

证明:由式(6)有:

$$y(k) = \frac{a_3 z^{-m-1}}{1 - a_1 z^{-1} - a_2 z^{-m-1}} u(k) \quad (21)$$

根据(20)式和(21)式画出采用 DMCAQM 算法的 TCP/AQM 系统的框图如图 1:

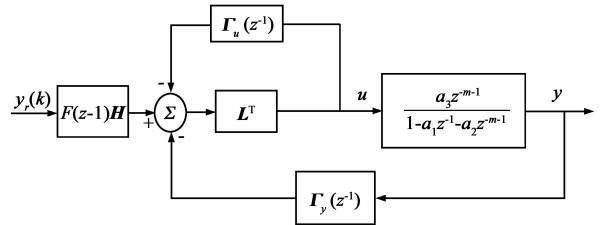


图 1 采用 DMCAQM 的 TCP/AQM 系统框图

化简框图,可知系统的特征方程为:

$$\begin{aligned} T(z^{-1}) &= (1 + \mathbf{L}^T \mathbf{\Gamma}_u(z^{-1})) (1 - a_1 z^{-1} \\ &\quad - a_2 z^{-m-1}) + a_3 z^{-m-1} \mathbf{L}^T \mathbf{\Gamma}_y(z^{-1}) \end{aligned} \quad (22)$$

所以,系统稳定的充要条件是 $T(z^{-1})$ 的所有解都在单位圆内.

3.2 参数选择

DMCAQM 算法需要设定的参数有采样间隔 dt , 预测步长 n , 误差加权矩阵 \mathbf{Q} , 控制作用加权矩阵 \mathbf{S} , 校正向量 \mathbf{H} , 目标队列长度 q_{ref} , 以及由网络环境相关参数 a_1, a_2, a_3 . 首先,根据网络环境,设置 a_1, a_2, a_3 . 如果它

们的取值与真实的网络参数很接近,控制效果会更好.但即使它们的设定值与实际值有偏差,由于算法的自校正设计,这部分偏差影响不大.在后面的仿真实验中,不同的网络环境用相同的控制器参数,DMCAQM 依然有效.

目标队列长度 q_{ref} 是缓冲队列稳定时的队列长度,可以根据用户需求设定.采样间隔 dt 是关键参数之一, dt 不能大于 R_0 的估计值,否则在确定预测模型的各项系数时, $m < 1$ 是不合理的.当然,采样间隔也不能过小.假如采样间隔取得过小, m 的值过大,会增加预测模型系数确定过程的计算量,同时测量的一个采样间隔的数据包达到数波动增大,不确定性增强. m 的理论取值范围为 $m \geq 1$.预测步长 n 理论上可以取大于 1 的任意正整数,但是 n 越大,预测模型系数确定过程的计算量越大.而且网络情况的变化很快,预测时域过长意义不大. n 的一般取值范围是 $2 \leq n \leq m$.校正向量 H 可以影响队列长度收敛的速度, H 越大队列收敛越快,队列长度的抖动也越大,其取值范围为 $H > 0$.最后误差加权矩阵 Q 和控制作用加权矩阵 S 的选择必须满足定理 1 所述的系统的稳定条件,以保障系统的稳定性.

4 仿真研究

仿下面将使用 NS2 网络仿真器进行不同网络环境下的仿真实验,以验证 DMCAQM 算法的有效性和优越性.如果没有特殊说明,使用的网络拓扑结构是如图 2 所示的哑铃状网络,其中路由器 B 和路由器 C 之间的链路为瓶颈链路.如不做特殊说明,使用的网络参数如下:瓶颈链路的容量为 45Mb/s,每个 TCP 连接的往返传

输时延为 0.1s,缓冲队列容量为 1125 个数据包 (packets),TCP 数据包的大小为 1000B,即 1packets = 1000B, TCP 连接数为 500 个. TCP 算法采用 TCP Reno,处理数据包的方式为包标记 (ENC).

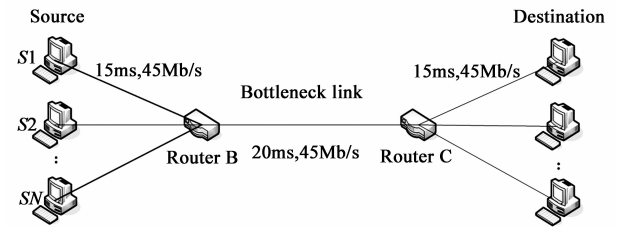


图2 单瓶颈链路的网络拓扑结构

DMCAQM 使用的参数如下: $a_1 = 0.9911$, $a_2 = -0.0089$, $a_3 = -28.2227$. 目标队列长度 $q_{ref} = 500$ packets, 采样间隔为 0.1s, $m = 2$, 预测步长 $n = 2$, 误差加权矩阵 $Q = \text{diag}(1, 0.8)$, 控制作用加权矩阵 $S = \text{diag}(10, 10)$, 校正向量 $H = \text{diag}(5, 0)$.

4.1 不同 TCP 链接数下的算法性能分析

在这组实验中,我们将观察不同 TCP 连接数量下的 DMCAQM 算法的有效性,仿真结果如图 3 所示.其中图 3(a)至(d)分别是 TCP 连接数为 50、100、500、1500 的情况下的队列长度变化的情况.可以看出,无论是轻载还是重载 DMCAQM 都能有效的控制队列长度,使其稳定在期望值.

4.2 不同往返时延

这组实验是为了观察 DMCAQM 在不同的往返时延中是否都有效,实验结果如图 4 所示.其中图 4(a)和图 4(b)是往返传输时延分别为 0.2s 和 0.05s 下的队列长

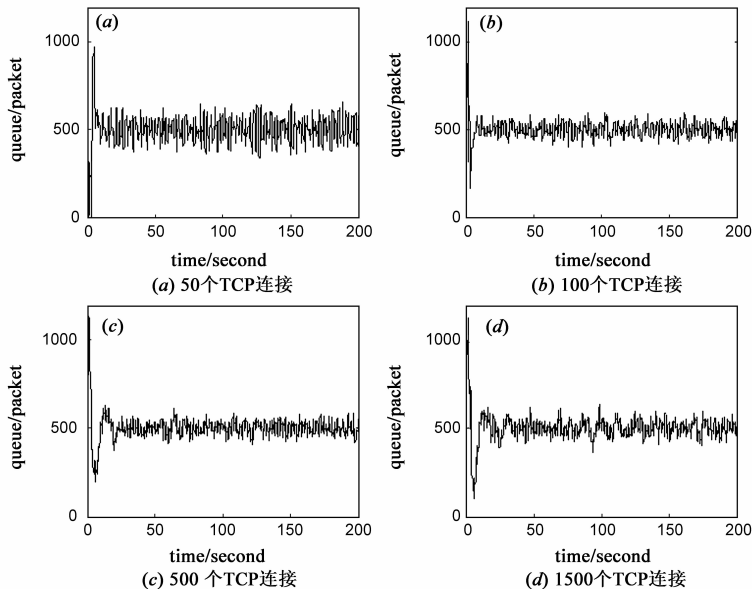


图3 不同TCP连接数量下的队列长度

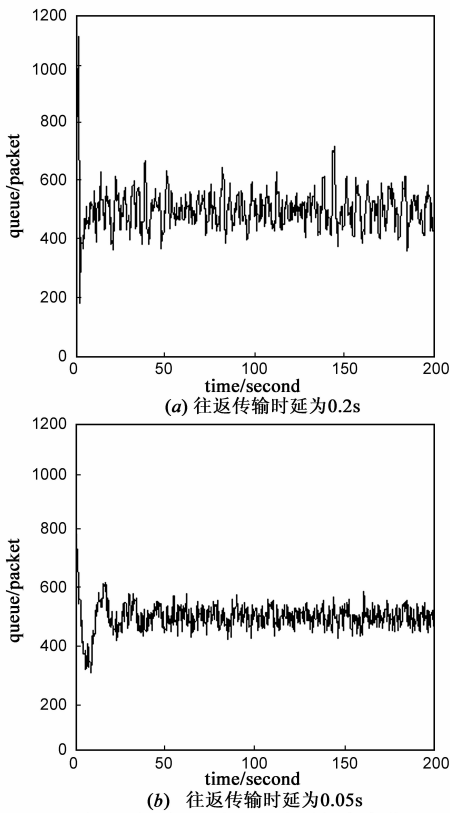


图4 不同往返时延下队列长度变化

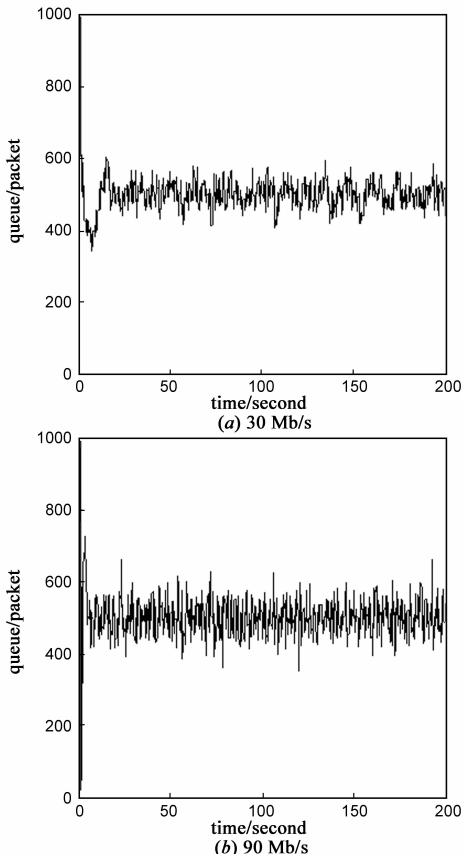


图5 不同瓶颈链路容量时队列长度的变化情况

度变化,结果表明,在不同的往返时延下,DMC-AQM能够有效的控制队列长度.

4.3 不同瓶颈链路容量

在这一组实验中,我们将观察不同瓶颈链路容量下 DMCAQM 算法是否有效,实验结果如图 5 所示,其中图 5(a)和图 5(b)分别为瓶颈链路容量为 30Mb 和 90Mb 时的队列变化情况.由图可知,在不同的瓶颈链路容量下 MACAQM 算法是有效的.

4.4 不同期望队列长度

这组实验是为了验证 DMCAQM 算法是否能够将队列长度稳定在设定的期望值,实验结果如图 6 所示,其中图 6(a)和图 6(b)分别是将期望队列长度设为 300packets 和 800packets 的情况下队列长度的变化.由实验结果可以看出,DMCAQM 可以将队列长度稳定到设定的期望值.

4.5 变化负载

在实际的网络中,TCP 连接数是时刻变化着的.在这组实验中,我们将分别观察 TCP 连接数缓慢变化和剧烈变化下 DMCAQM 的表现.

(1)TCP 连接数缓慢变化的仿真设置如下:起始时有 500 个 TCP 连接,第 30s 开始,每 0.1s 增加 1 个 TCP

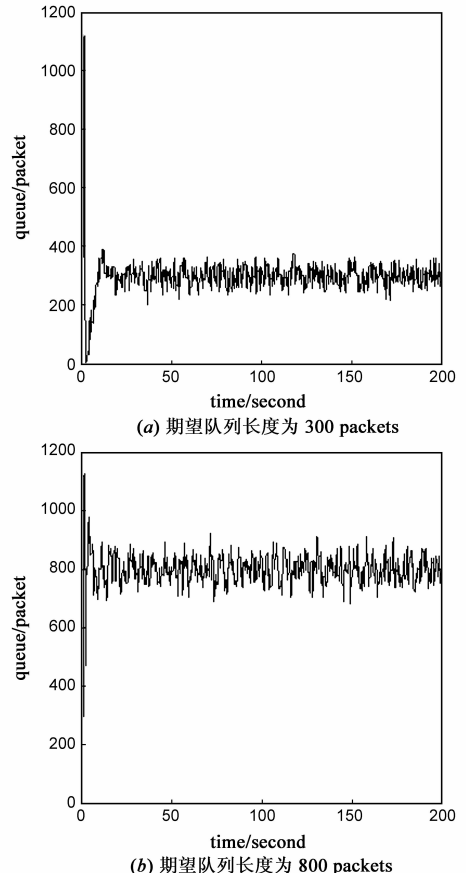


图6 不同期望队列长度的队列长度变化

连接, 总共增加 500 个, 第 120s 开始, 每 0.1s 减少一个, 总共减少 500 个. 实验结果如图 7 所示, 其中图 7(a) 是队列长度的变化, 图 7(b) 是丢弃概率的变化. 由图 7 可知, 随着 TCP 连接数的缓慢增加或减少, 丢弃概率也缓慢增大或减小, 但是队列长度稳定在期望值附近. 这说明即使 TCP 连接数缓慢变化, DMCAQM 算法仍然是有效的.

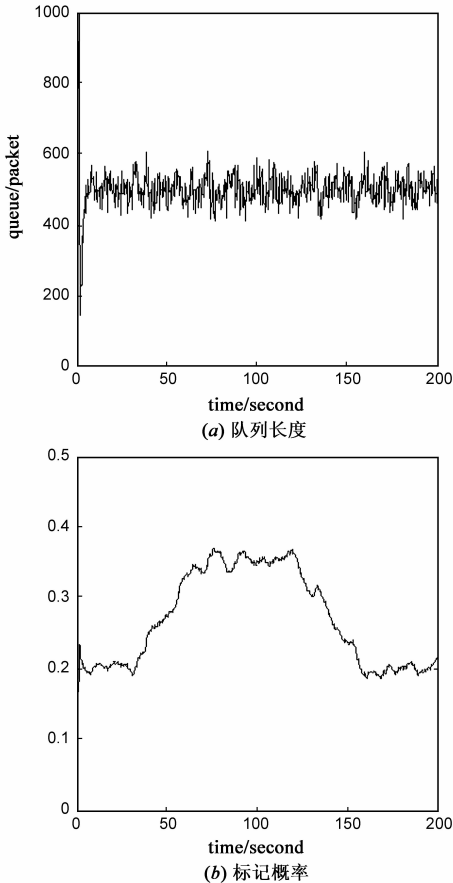


图7 TCP数目缓慢变化时队列长度和标记概率的变化

(2) TCP 连接数剧烈变化的仿真设置如下: 起始时有 500 个 TCP 连接, 第 30s、50s、70s 时突然增加 100 个 TCP 连接, 而在第 120s、140s、160s 时突然减少 100 个 TCP 连接. 实验结果如图 8 所示, 其中图 8(a) 是队列长度的变化, 图 8(b) 是丢弃概率的变化. 由图 8 可知, 如果 TCP 连接数突然增加, 丢弃概率随着突然增大, 经过一定时间后稳定, 队列长度在经过一定时间后也稳定在期望值; 如果 TCP 连接数突然减少, 丢弃概率随着突然减小, 队列长度在经过一段时期后也稳定在期望值. 但是在 TCP 连接数突然变化的时刻, 队列长度会有抖动, 这是由于控制量(数据包丢弃概率)需要随系统环境的变化而改变, 使系统在工作在新的稳定点附近, 而这个过程需要一定的过度时间. 如何缩短这个过度时间, 也是 DMCAQM 算法今后需要改进的地方. 无论如

何, 由实验结果可以看出, 即使 TCP 连接数剧烈变化, DMCAQM 算法仍然是有效的.

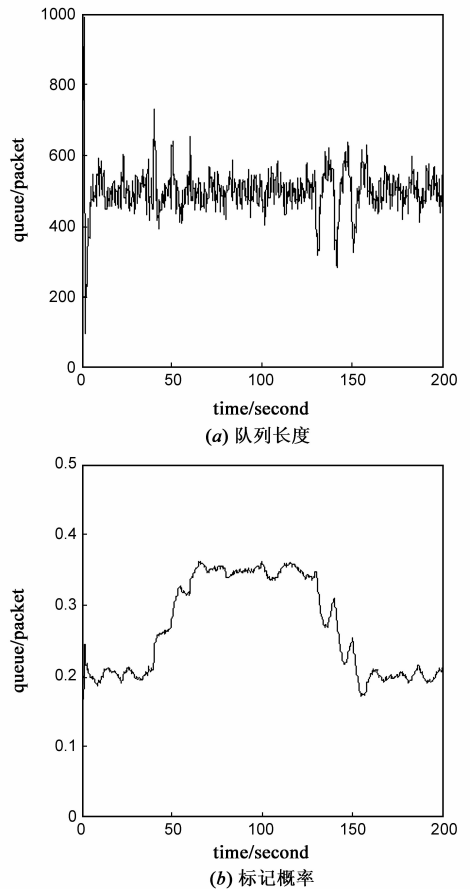


图8 TCP数目剧烈变化时队列长度和标记概率的变化

4.6 多瓶颈链路

为了观察 DMCAQM 在多瓶颈链路下的控制效果, 本节将给出多瓶颈链路网络的实验结果. 设置的多瓶颈链路如图 9 所示, 其中 R1 和 R2、R3 和 R4 之间的链路为瓶颈链路. 共有 300 个 TCP 连接的源端连接在 R1 上, 其中各有 100 个 TCP 连接的目的端分别连接在 R2、R3 和 R4 上. 另外还有 200 个 TCP 连接的源端连接在 R2 上, 其中各有 100 个 TCP 连接的目的端分别连接在 R3 和 R4. 最后还有 100 个 TCP 连接由 R3 到达 R4. 实验结果如图 10 所示, 其中图 10(a) 和图 10(b) 分别是 R1 和 R2、R3 和 R4 之间的瓶颈链路的队列变化情况. 可以看出, 在复杂网络中 DMCAQM 也可以有效的控制

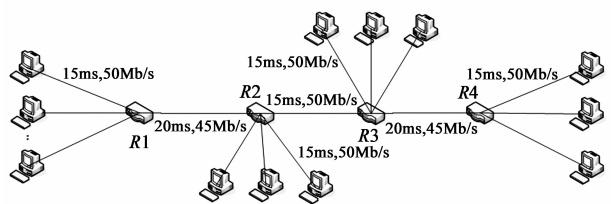


图9 多瓶颈链路的拓扑结构

队列长度,使其稳定在设定的期望值附近.

4.7 与其他算法比较

这组实验将 DMCAQM 与 PI、REM 及 RaQ 等算法进行比较.使用的网络结构是图 2 所示的哑铃状网络,DMCAQM 采用前面描述的默认设置.为了使实验结果更加明显,仿真时间缩减为 100s.PI 和 REM 使用 NS2 中的默认参数,RaQ 使用的参数如下: $q_{hp} = 0.0077$, $q_{ki} = 0.0005$, $r_{hp} = 0.0095$,采用间隔为 0.005s.值得注意的是,DMCAQM 使用的采样间隔为 0.1s,远远大于其他算法的采样间隔,这意味着大大减少了路由器的计算量,节省资源.如果其他算法的采样间隔设为 0.1s,其队列

长度需要更长的时间稳定,队列抖动更大.

实验结果如图 11 所示,可以看出 PI 和 REM 的收敛速度最慢,RaQ 算法收敛速度相对于前两个算法要快一些,但是稳定之前经过了几次大幅度的震荡.而 DMCAQM 是其中收敛速度最快,队列抖动最小的算法.

5 结论

本文通过对流体流模型的理论分析,提出了新的预测模型,该模型形式简单,参数计算相对容易,并能够根据当前的网络情况有效的预测拥塞窗口的变化.本文利用这个模型,并结合 DMC 的思想,提出了一个有

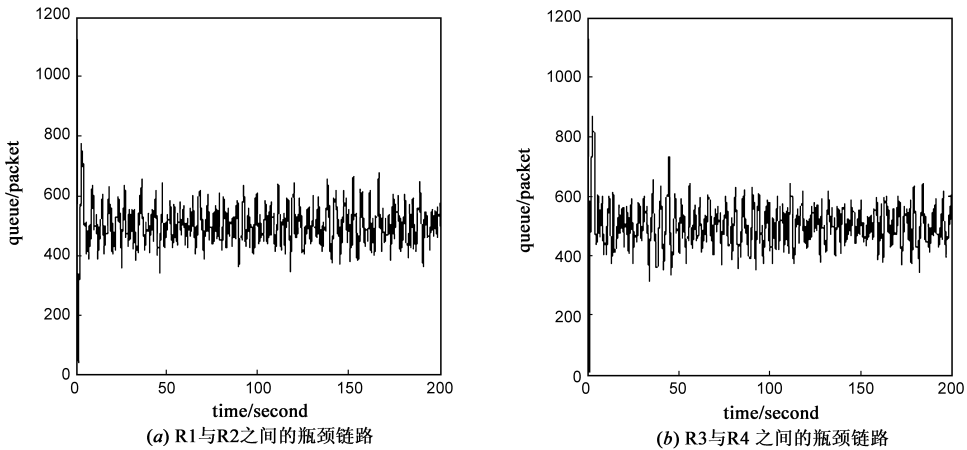


图10 复杂网络队列长度变化

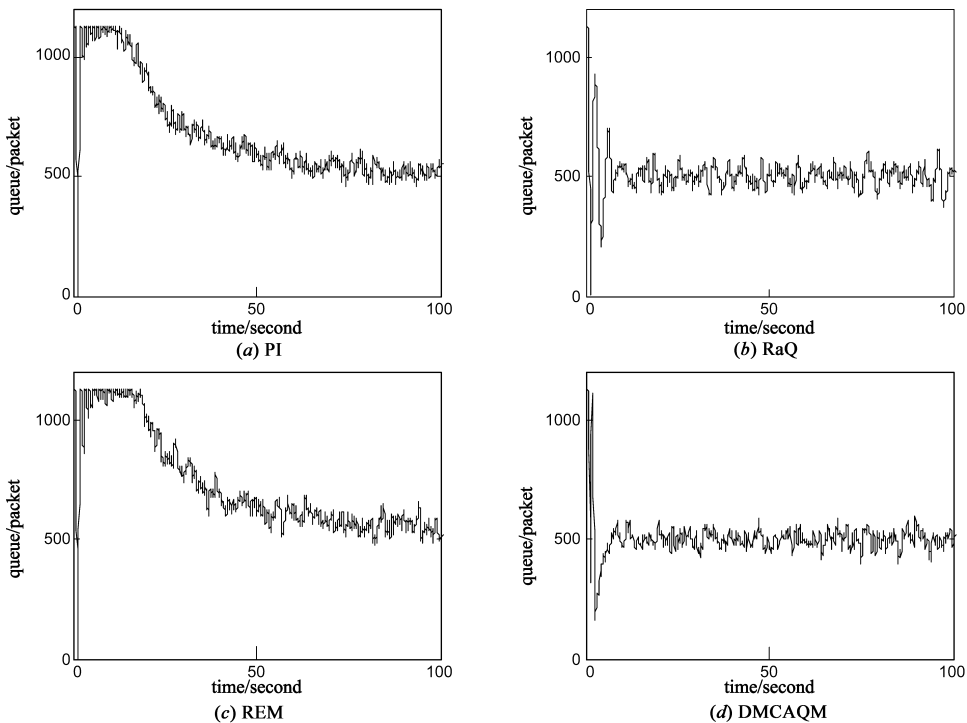


图11 不同算法之间的比较

效的主动队列管理算法 DMCAQM. 该算法具有较好的鲁棒性, 与传统的算法相比, 占用的资源更少, 并且有更快的收敛速度和更小的队列抖动.

参考文献

- [1] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance[J]. IEEE/ACM Transactions on Networking, 1993, 1(4): 397 – 413.
- [2] Kunniyur S, Srikant R. An analysis and design of an Adaptive Virtual Queue (AVQ) algorithm for active queue management [A]. Proceedings of the 2001 conference on Applications, Technologies, Architectures, and Protocols for Computer Communications[C]. New York: ACM, 2001. 31(4): 123 – 134.
- [3] Misra V, Gong W B, Towsley D. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED[A]. Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication[C]. New York: ACM, 2000. 30(4): 151 – 160.
- [4] 陆锦军, 王执铨. 基于粒子群优化的网络拥塞控制新算法[J]. 电子学报, 2007, 35(8): 1446 – 1451.
Lu Jin-jun, Wang Zhi-quan. A new network congestion control algorithm based on particle swarm optimization[J]. Acta Electronica Sinica, 2007, 35(8): 1446 – 1451. (in Chinese)
- [5] 葛龙, 孙金生, 王执铨. 基于动态输出反馈控制的主动队列管理算法[J]. 电子学报, 2010, 38(12): 2816 – 2821.
Ge Long, Sun Jin-sheng, Wang Zhi-quan. Active queue management algorithm based on dynamic output feedback control [J]. Acta Electronica Sinica, 2010, 38(12): 2816 – 2821. (in Chinese).
- [6] Bigdeli N, Haeri M. Predictive functional control for active queue management in congested TCP/IP networks [J]. ISA Transactions, 2009, 48(1): 107 – 121.
- [7] MARAMI B, HAERI M. Implementation of MPC as an AQM controller[J]. Computer Communications, 2010, 33(2): 227 – 239.

- [8] Wang P, Chen H, Lu X H. Dynamic matrix control for active queue management based on step response model[A]. Proceedings of the 8th World Congress on Intelligent Control and Automation[C]. New York: IEEE, 2010. 1093 – 1097.
- [9] Wang P, Chen H, Yang X P, Ma Y. Design and analysis of a model predictive controller for active queue management[J]. ISA Transactions, 2012, 51(1): 120 – 131.
- [10] Hollot C V, Misra V, Towsley D, Gong W B. A control theoretic analysis of RED[A]. INFOCOM 2001 Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies[C]. New York: IEEE, 2001. 3: 1510 – 1519.
- [11] Clarke D W, Mohtadi C, Tuffs P S. Generalized predictive control-part I. the basic algorithm[J]. Automatica, 1987, 23(2): 137 – 148.

作者简介



徐 琴 女, 1987 年出生, 江西丰城人. 2009 年毕业于南京理工大学自动化学院电子信息工程专业, 现为南京理工大学自动化学院博士研究生. 主要研究方向为网络拥塞控制.

E-mail: missxuqin@126.com



孙金生 男, 1967 年出生, 吉林伊通人. 教授、博士生导师. 1990 年、1993 年和 1995 年分别在南京理工大学获工学学士、工学硕士和工学博士学位. 主要从事网络拥塞控制、质量控制等方面的研究工作.

E-mail: jssun67@163.com