

# TCP-RED 离散反馈系统中的边界碰撞 分岔及混沌控制<sup>\*</sup>

杨 谈<sup>†</sup> 金跃辉 程时端

(北京邮电大学网络与交换技术国家重点实验室, 北京 100876)

(2009 年 3 月 17 日收到, 2009 年 5 月 21 日收到修改稿)

结合范式方法对传输控制协议和随机早期检测(TCP-RED)离散反馈系统中出现的不同种类的边界碰撞分岔现象进行了分析. 随着系统参数的变化, 系统将经历边界碰撞分岔并从周期稳态向混沌态跃变, 导致系统中瞬时队列的长度出现剧烈振荡, 严重影响系统性能. 根据不动点的线性稳定条件, 提出了一种混沌控制方法, 通过对系统不动点邻域内的状态变量进行扰动, 将系统引导至不动点. 该方法能显著提高 TCP-RED 系统的性能和资源利用率. 仿真结果证明了该方法的有效性.

关键词: 传输控制协议, 随机早期检测, 边界碰撞分岔, 混沌控制

PACC: 0545

## 1. 引 言

传输控制协议(TCP)和随机早期检测(RED)算法目前广泛应用于互联网的拥塞控制. RED 算法使用平均队列长度度量网络的拥塞程度, 并依此对分组进行概率丢弃; 而 TCP 基于丢包监测机制来判断是否启动拥塞控制策略, 并通过降低数据传输速率进行拥塞控制.

RED 和 TCP 机制的结合是目前网络拥塞控制的主要途径. 文献 [1] 中将 TCP 拥塞控制和 RED 队列管理模型简化成一个离散反馈系统, 以系统的平均队列长度为状态变量, 得到系统的分段光滑的动力学方程; 同时指出随着控制参数的变化系统中将会出现倍周期分岔, 边界碰撞分岔以及混沌等现象.

边界碰撞分岔<sup>[2]</sup>是指分段光滑系统中随着控制参数的变化, 系统的不动点或某个周期点位于系统中的某条边界上(即系统周期轨道与边界碰撞), 从而导致系统的动力学行为发生剧烈变化的现象. 边界碰撞分岔包括系统从周期态跃变至混沌态, 混沌态跃变至周期态以及周期态间的跃变等多种形态. Banerjee 等将系统中边界邻域的分段映射局部线性

化, 通过对所得范式的线性化特征进行分类, 给出了边界碰撞分岔的分类方法<sup>[3,4]</sup>.

系统边界碰撞分岔现象在 TCP-RED 离散反馈系统中以多种形式大量出现, 特别是当系统发生从周期态向混沌态跃变时将导致队列长度的大幅度振荡, 网络中的端到端时延抖动增大, 对网络运行极其不利. 本文将结合边界碰撞分岔的已有分类对 TCP-RED 系统中的边界碰撞分岔现象进行深入讨论. 同时, 根据不动点的线性稳定条件提出了一种混沌控制方法, 通过对系统不动点邻域内的状态变量的扰动, 消除该区域系统的不稳定性, 最终将系统引导至不动点.

## 2. TCP-RED 离散反馈系统模型

考虑如图 1 所示的简单网络. 该网络中  $N$  个 TCP-Reno 连接共享一条容量为  $C$  的瓶颈链路, 且网络双向时延(不包括排队时延)为  $d$ .

TCP-RED 反馈控制机制如图 2 所示. TCP 发送端根据  $k$  时刻的丢包率  $p_k$  来调整下一时刻的发送速率, 该速率与路由器中的瞬时队列长度  $q_{k+1}$  密切

<sup>\*</sup> 国家重点基础研究发展计划(批准号: 2009CB320505, 2009CB320504)和国家高技术研究发展计划(批准号: 2006AA01Z235)资助的课题.

<sup>†</sup> E-mail: yatomoto@gmail.com

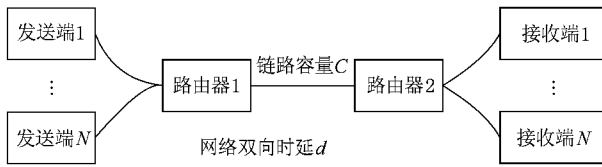


图 1 网络拓扑

相关，即有  $q_{k+1} = \alpha(p_k)$ ；而路由器中的 RED 控制器基于  $q_{k+1}$  通过指数加权法计算得到平均队列长度  $\bar{q}_{k+1} = A(q_{k+1}, \bar{q}_k)$ ，并根据  $\bar{q}_{k+1}$  调整丢包概率  $p_{k+1} = H(\bar{q}_{k+1})$ ，反馈至 TCP 发送端。

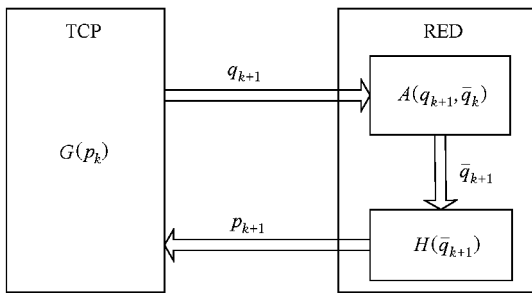


图 2 TCP-RED 反馈控制机制

在网络双向延迟  $d$  和丢包率  $p_k$  已知的情况下，单个 TCP-Reno 连接的吞吐量可近似为<sup>[5-7]</sup>

$$T(p_k, d) \approx \frac{MK}{\sqrt{p_k d}}, \quad (1)$$

其中  $M$  为平均包长， $K$  为取值在  $[1, \sqrt{8/3}]$  区间内

的常数。由于网络吞吐量不大于  $C$ ，则由

$$NT(p_k, d) = N \frac{MK}{\sqrt{p_k d}} = C \quad (2)$$

可求得系统在稳定状态下的最大丢包率  $p_u = (NMK/dC)^2$ ，若  $p_k \geq p_u$ ，TCP 启动拥塞控制，降低数据的传输速率，队列在下一时刻将为空，即  $q_{k+1} = 0$ ；而当  $p_k < p_u$  时， $q_{k+1} \neq 0$ ，有

$$NT(p_k, d) = N \frac{MK}{\sqrt{p_k} \left( d + \frac{q_{k+1} M}{C} \right)} = C, \quad (3)$$

因而可得

$$q_{k+1} = \frac{NK}{\sqrt{p_k}} - \frac{Cd}{M}. \quad (4)$$

根据 (4) 式即可求得  $q_{k+1}$  等于队列缓冲量  $B$  时的最大丢包率  $p_l = (NMK/(dC + BM))^2$ 。

由上可知，网络路由器中瞬时队列长度  $q_{k+1}$  与丢包率  $p_k$  的关系式为

$$q_{k+1} = \alpha(p_k) = \begin{cases} 0 & (p_k \geq p_u), \\ B & (p_k \leq p_l), \\ \frac{NK}{\sqrt{p_k}} - \frac{Cd}{M} & (p_l < p_k < p_u). \end{cases} \quad (5)$$

同时 RED 队列管理机制运用指数加权法计算得到平均队列长度

$$\bar{q}_{k+1} = A(q_{k+1}, \bar{q}_k) = (1-w)\bar{q}_k + wq_{k+1}, \quad (6)$$

其中  $w$  为权重系数。此时 RED 将根据计算得到的平均队列长度调整丢包概率如下：

$$p_{k+1} = H(\bar{q}_{k+1}) = \begin{cases} 0 & (\bar{q}_{k+1} < q_{\min}), \\ 1 & (\bar{q}_{k+1} > q_{\max}), \\ \frac{\bar{q}_{k+1} - q_{\min}}{q_{\max} - q_{\min}} p_{\max} & (q_{\min} \leq \bar{q}_{k+1} \leq q_{\max}), \end{cases} \quad (7)$$

其中  $q_{\min}$  和  $q_{\max}$  分别表示最小和最大队列长度， $p_{\max}$  为最大丢包概率。

由 (5)–(7) 式得到以平均队列长度为状态变量的 TCP-RED 离散反馈系统分段光滑的动力方程如下：

$$\bar{q}_{k+1} = f(\bar{q}_k, p_k) = \begin{cases} (1-w)\bar{q}_k & (\bar{q}_k \geq \bar{q}_u), \\ (1-w)\bar{q}_k + wB & (\bar{q}_k \leq \bar{q}_l), \\ (1-w)\bar{q}_k + w \left( \frac{NK}{\sqrt{p_k}} - \frac{Cd}{M} \right) & (\bar{q}_l < \bar{q}_k < \bar{q}_u), \end{cases} \quad (8)$$

其中  $\rho$  表示系统参数, 包括权重系数  $w$ , TCP 连接数  $N$ , 最小队列长度  $q_{\min}$ , 最大队列长度  $q_{\max}$  以及最大丢包概率  $p_{\max}$ .  $\bar{q}_u$  和  $\bar{q}_l$  分别对应丢包率为  $p_u$  和  $p_l$  时的平均队列长度, 可由 (7) 式计算得到.

### 3. TCP-RED 系统中边界碰撞分岔现象的分析

较之常见的倍周期分岔, 边界碰撞分岔仅出现在分段光滑的非线性动力系统中<sup>[8-12]</sup>. 本文中只讨论 TCP-RED 系统的边界碰撞分岔.

图 3 中给出了分别以权重系数  $w$ 、最小队列长度  $q_{\min}$ 、最大队列长度  $q_{\max}$ 、最大丢包概率  $p_{\max}$ 、TCP 连接数  $N$ 、链路带宽  $C$  以及网络双向时延  $d$  为分岔参数, 数值仿真得到的状态变量  $\bar{q}_k$  的分岔图. 其他参数值选取如下:  $K = \sqrt{3/2}$ ,  $B = 3750$  包,  $M = 4000$  bit. 以  $\bar{q}_0 = 0$  为初值迭代 1000 次, 将最后的 200 个迭代值示于分岔图中. 同时与  $\bar{q}_k$  对应的瞬时队列长度  $q_k$  的状态如图 4 所示.

从图 3 和图 4 可以看到, 当以不同参数作为分岔参数时, TCP-RED 系统中存在着相似的分岔现象. 由于篇幅所限, 本文中以分岔参数取 TCP 连接数  $N$  为例讨论 TCP-RED 系统中的边界碰撞分岔.

首先介绍边界碰撞分岔中常用的范式方法. 范式方法即指系统动力方程  $f(\cdot)$  迭代  $n$  次后, 将所得迭代方程  $f^{(n)}$  在边界邻域的分段映射局部线性化, 得如下范式<sup>[3]</sup>:

$$F^{(n)}(x', \mu) = \begin{cases} a^{(n)}x' + \mu & (x' \leq 0), \\ b^{(n)}x' + \mu & (x' \geq 0), \end{cases} \quad (9)$$

其中  $x'$  为系统中将边界  $x_B$  平移至 0 后的状态变量, 即  $x' = x - x_B$ .  $\mu$  为分岔参数(系统中的某个参数). 为方便讨论, 假设  $\mu = 0$  时发生边界碰撞分岔, 有

$$a^{(n)} = \lim_{x \rightarrow 0^-} \frac{\partial}{\partial x} f^{(n)}(x, 0),$$

$$b^{(n)} = \lim_{x \rightarrow 0^+} \frac{\partial}{\partial x} f^{(n)}(x, 0).$$

$a^{(n)}$  和  $b^{(n)}$  的值反映了动力方程在边界邻域的局部特征, 同时决定了边界碰撞分岔的性质. 故通过分析系统对应的范式, 即可得到 TCP-RED 系统中边界碰

撞分岔的原因和种类.

图 3(e) 给出了  $N$  在 100—300 范围时平均队列长度  $\bar{q}_k$  的混沌分岔图. 随着  $N$  的增加, 出现了三种边界碰撞分岔方式.

第一种方式是周期态  $\rightarrow$  周期态的边界碰撞. 当  $N = 140$  时, 周期 8 轨道穿过边界 1, 系统出现碰撞分岔, 从周期 8 轨道进入另一个周期 8 轨道, 如图 3(a) 所示. 边界 1 邻域内  $f^{(8)}$  的范式形式如图 6(a) 所示, 其中  $-1 < a^{(8)} \leq b^{(8)} < 1$ . 当  $N > 140$  后  $f^{(8)}$  上的一个稳定不动点变为另一个稳定不动点, 导致系统从一种 8 周期稳定态转换为另一种 8 周期稳定态.

第二种方式是周期态  $\rightarrow$  混沌态的边界碰撞. 当  $N = 146$  时, 周期 8 轨道穿过边界 2, 系统经历边界碰撞分岔从 8 周期运动进入混沌, 如图 3(a) 所示. 边界 2 邻域内  $f^{(8)}$  的范式形式如图 6(b) 所示, 其中  $-1 < a^{(8)} < 1 < b^{(8)}$ . 当  $N > 146$  后  $f^{(8)}$  从有一个稳定不动点变为没有不动点的存在, 导致系统从 8 周期态跃变为混沌态.

第三种方式是混沌态  $\rightarrow$  周期态的边界碰撞. 当  $N = 237$  时, 周期 2 轨道穿过边界 1, 系统经过边界碰撞分岔进入 2 周期运动, 如图 3(b) 所示. 边界 1 邻域内  $f^{(2)}$  的范式形式如图 6(c) 所示, 其中  $0 < a^{(2)} < 1, b^{(2)} < -1$ . 当  $N > 237$  后  $f^{(2)}$  从有一个不稳定不动点变为有一个稳定不动点, 导致系统从混沌态跃变为 2 周期态.

通过以上分析可以看到, 当网络中 TCP 连接较少时, 网络的平均队列长度  $\bar{q}_k$  处于稳定的周期态, 当 TCP 连接数增加到某个值时, 系统将经历边界碰撞分岔, 从而突然进入混沌态. 而随着 TCP 连接数的继续增加, 系统将再次经历边界碰撞分岔, 平均队列长度最终达到稳态. 同时分析图 4(e) 可以看到, 与平均队列长度  $\bar{q}_k$  相对应的瞬时队列长度  $q_k$  (即路由器中的真实队列长度) 经历了与  $\bar{q}_k$  类似的情况, 从混沌态最终进入稳态. 瞬时队列长度与网络排队时延密切相关, 瞬时队列长度的混沌态将增大网络端到端时延抖动, 严重影响网络性能; 且当 TCP 连接数较少时, 图中显示  $q_k$  值为零, 说明此时系统资源利用率极低.

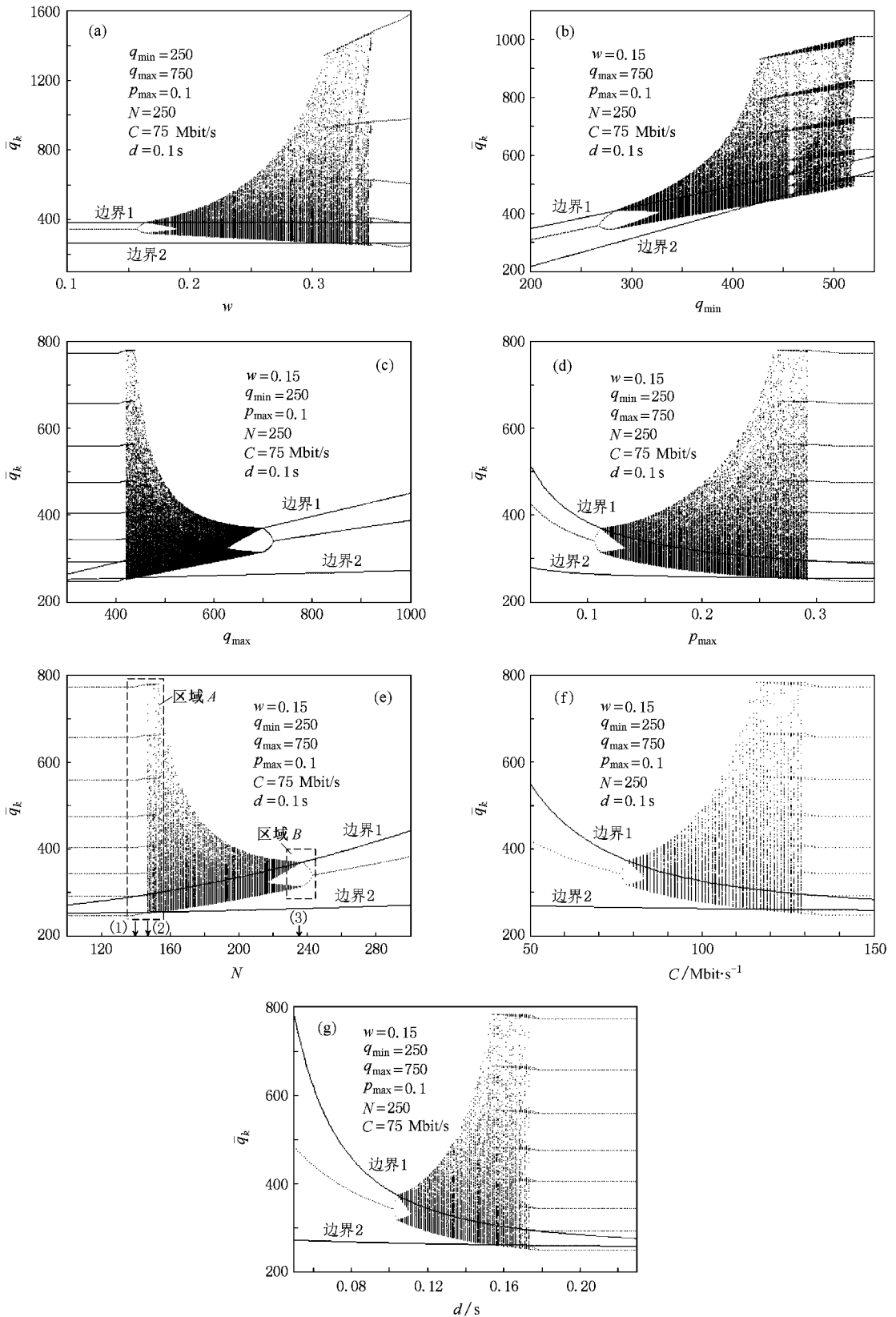


图3 TCP-RED系统中  $\bar{q}_k$  的分岔图 (a)  $w$  为分岔参数, (b)  $q_{\min}$  为分岔参数, (c)  $q_{\max}$  为分岔参数, (d)  $p_{\max}$  为分岔参数, (e)  $N$  为分岔参数, (f)  $C$  为分岔参数, (g)  $d$  为分岔参数

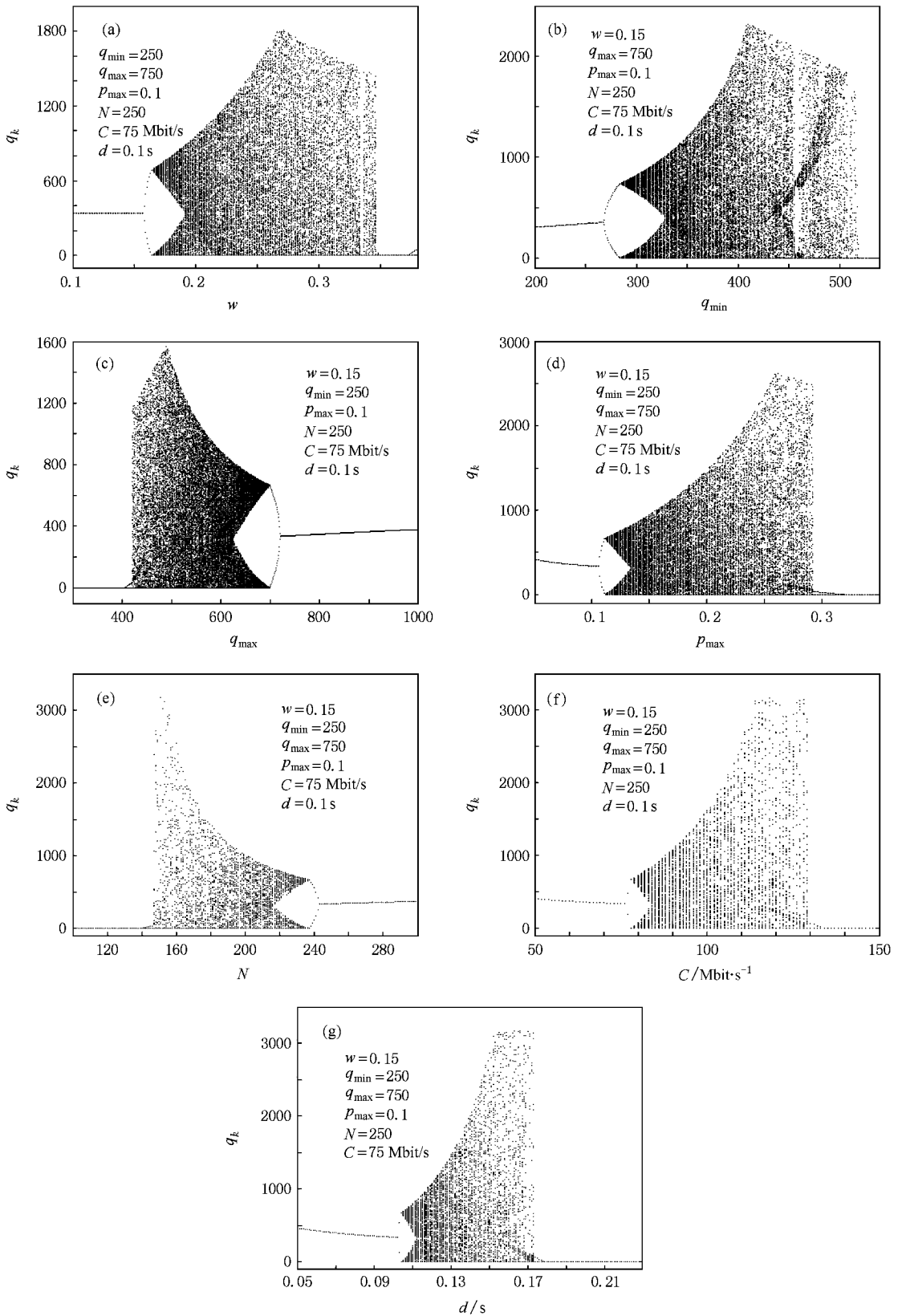


图4 TCP-RED系统中  $q_k$  的分岔图 (a)  $w$  为分岔参数, (b)  $q_{\min}$  为分岔参数, (c)  $q_{\max}$  为分岔参数, (d)  $p_{\max}$  为分岔参数, (e)  $N$  为分岔参数, (f)  $C$  为分岔参数, (g)  $d$  为分岔参数

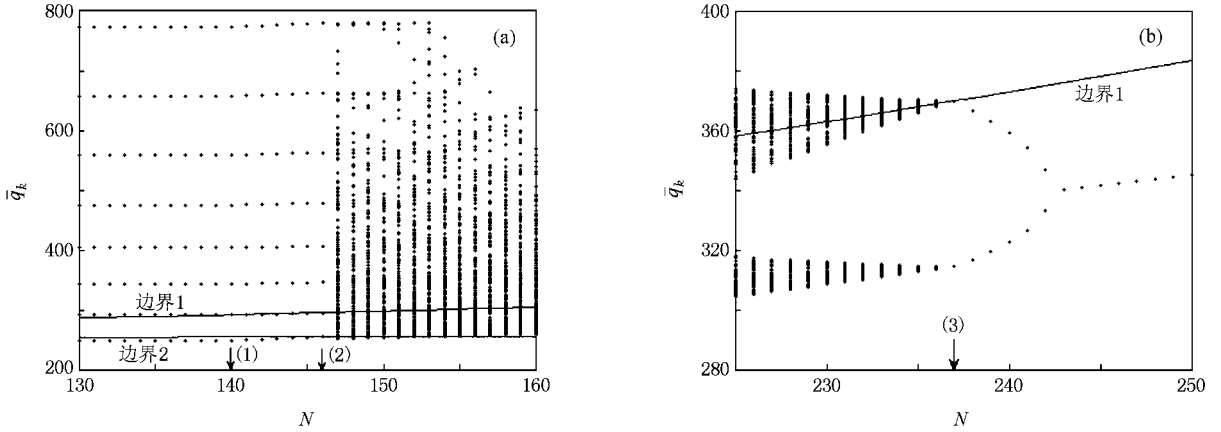


图 5 TCP-RED 系统中放大的  $\bar{q}_k$  分岔图 (a)图 3(e)中的区域 A,  $N = 140$  和  $N = 146$ , (b)图 3(e)中的区域 B,  $N = 247$

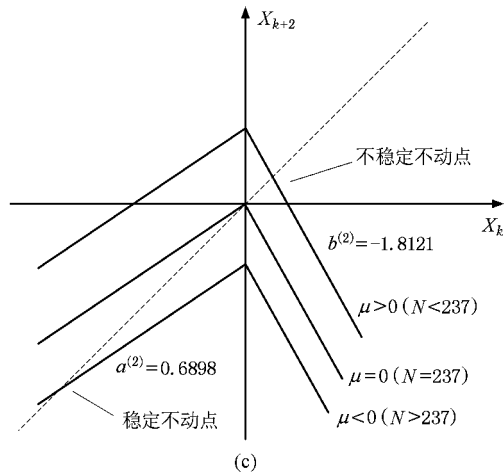
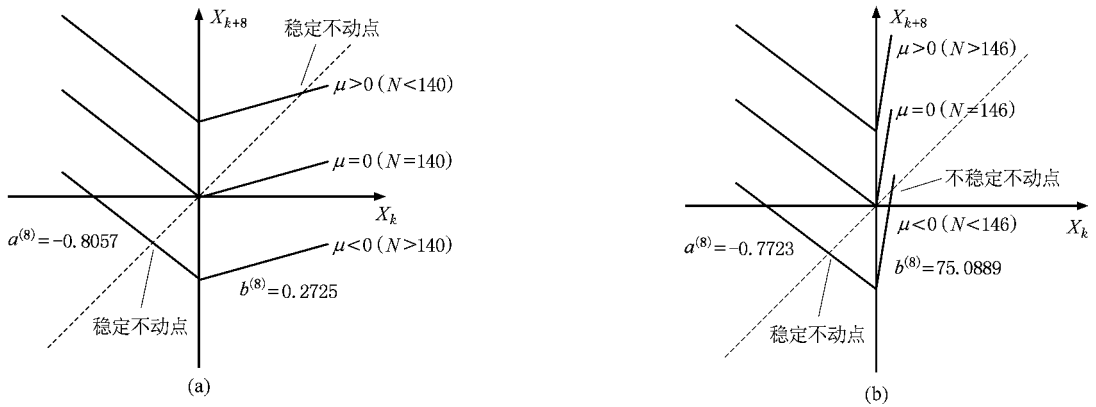


图 6 范式示意图 (a)  $f^{(8)}$  在  $N = 140$  附近, (b)  $f^{(8)}$  在  $N = 146$  附近, (c)  $f^{(2)}$  在  $N = 237$  附近

## 4. TCP-RED 系统中混沌控制

当 TCP-RED 系统从周期态向混沌态跃变时,系统中的瞬时队列长度将大幅度振荡,导致网络端到端时延抖动增大. 目前对 TCP-RED 系统中的混沌控制研究不多. 文献 [13, 14] 中运用混沌控制中常见的延迟反馈控制法对 TCP-RED 系统进行控制,通过对系统中的  $p_{\max}$  参数进行扰动达到最终稳定系统的目标,但该方法中反馈系数取值的非自适应性降低了其在实际系统中的实用性. 文献 [15] 提出了一种控制方法,当系统检测到混沌发生<sup>[13, 14]</sup>后,直接把系统中的平均队列长度指定为不动点,该方法简单有效,但完全破坏了 RED 的控制机制.

### 4.1. 混沌控制方法

本文根据不动点的稳定条件提出了一种混沌控制的方法. 通过对不动点邻域内的状态变量  $\bar{q}_k$  进行扰动,消除该区域系统的不稳定性,将系统有效稳定在不动点,达到混沌控制.

设系统 (8) 中平均队列长度的不动点为  $\bar{q}_k^*$ , 即  $\bar{q}_k^* = \mathcal{K}(\bar{q}_k^*, \rho)$ . 由 (7) 和 (8) 式得

$$\bar{q}_k^* = \frac{NK}{\sqrt{p_k^*}} - \frac{Cd}{M}, \quad (10)$$

其中  $p_k^*$  为方程

$$\frac{CM(q_{\max} - q_{\min})y^3}{p_{\max}} + (CMq_{\min} + dC^2)y - NMKC = 0 \quad (11)$$

的正实根的正平方.

系统 (8) 在  $\bar{q}_k^*$  的一阶导数  $S$  为

$$S = \left. \frac{\partial}{\partial q_k} \mathcal{K}(\bar{q}_k, \rho) \right|_{\bar{q}_k^*} = 1 - w - \frac{wNK}{2p_{\max}^{1/2}(q_{\max} - q_{\min})^{1/2}(\bar{q}_k^* - q_{\min})^{3/2}}, \quad (12)$$

则使不动点线性稳定的条件为

$$|S| < 1. \quad (13)$$

若要将系统稳定在不动点, 可对  $\bar{q}_k^*$  的邻域  $[\bar{q}_k^* - \epsilon, \bar{q}_k^* + \epsilon]$  ( $0 < \epsilon \ll \infty$ ) 内的状态变量  $\bar{q}_k$  进行扰动. 当  $\bar{q}_k$  位于该区域时, 在系统 (8) 中添加控

制项  $u_k$ , 使得

$$\begin{aligned} \bar{q}_{k+1} &= \mathcal{K}(\bar{q}_k, \rho) = \mathcal{K}(\bar{q}_k, \rho) + u_k \\ (\bar{q}_k^* - \epsilon &\leq \bar{q}_k \leq \bar{q}_k^* + \epsilon), \end{aligned} \quad (14)$$

且

$$u_k = \gamma(\bar{q}_k - \bar{q}_k^*), \quad (15)$$

其中  $\gamma$  为控制系数. 通过适当选取  $\gamma$  的值, 使得系统 (8) 在  $[\bar{q}_k^* - \epsilon, \bar{q}_k^* + \epsilon]$  内线性稳定, 即

$$\left| \left. \frac{\partial}{\partial q_k} \mathcal{K}(\bar{q}_k, \rho) \right|_{\bar{q}_k^*} \right| = |S + \gamma| < 1. \quad (16)$$

### 4.2. 数值仿真结果

分别以权重系数  $w$ 、最小队列长度  $q_{\min}$ 、最大队列长度  $q_{\max}$ 、最大丢包概率  $p_{\max}$ 、TCP 连接数  $N$ 、链路带宽  $C$  以及网络双向时延  $d$  为分岔参数, 使用以上控制方法对系统进行控制. 参数取值与上述相同. 由 (10—12) 式计算得到  $\bar{q}_k^*$ ,  $S$ , 并选取  $\epsilon = 25$ ,  $\gamma = -S - 0.5$ , 根据 (14) 和 (15) 式实施控制.  $\bar{q}_k$  的控制结果如图 7 所示, 实施控制后  $q_k$  的状态如图 8 所示.

从图 7 和图 8 可以看到, 通过对不动点邻域内的状态变量进行扰动, 系统中的平均队列长度以及瞬时队列长度被控制在稳态, 系统中的端到端时延抖动得到了有效遏制, 系统性能得到了改善. 同时对比图 4 和图 8 可以看到, 当系统中  $N$  和  $q_{\max}$  的值较小或者  $w, q_{\min}, p_{\max}, C, d$  的值较大时, 系统资源的利用率极低, 该现象和实际网络系统中的情况一致; 而实施控制后, 系统中的瞬时队列长度保持在稳定的水平, 系统资源利用率得到了明显提高.

### 4.3. NS2 仿真结果

由于 TCP 和 RED 均广泛应用于实际网络中, 故这里使用 NS2 网络仿真软件来验证该控制方法的有效性.

仿真场景的网络拓扑如图 1 所示, 发送端和接收端分别为 500 个, 其中第  $i$  ( $i = 1, 2, \dots, 500$ ) 个发送端和第  $i$  ( $i = 1, 2, \dots, 500$ ) 个接收端建立一个 TCP-Reno 连接, 故网络中的 TCP 连接数  $N = 500$ , 且设平均包大小  $M = 4000$  bit. 连接发送端和路由器 1 以及连接路由器 2 和接收端的边缘链路上的传播时延均服从  $[10 \text{ ms}, 35 \text{ ms}]$  区间上的均匀分布, 链路带

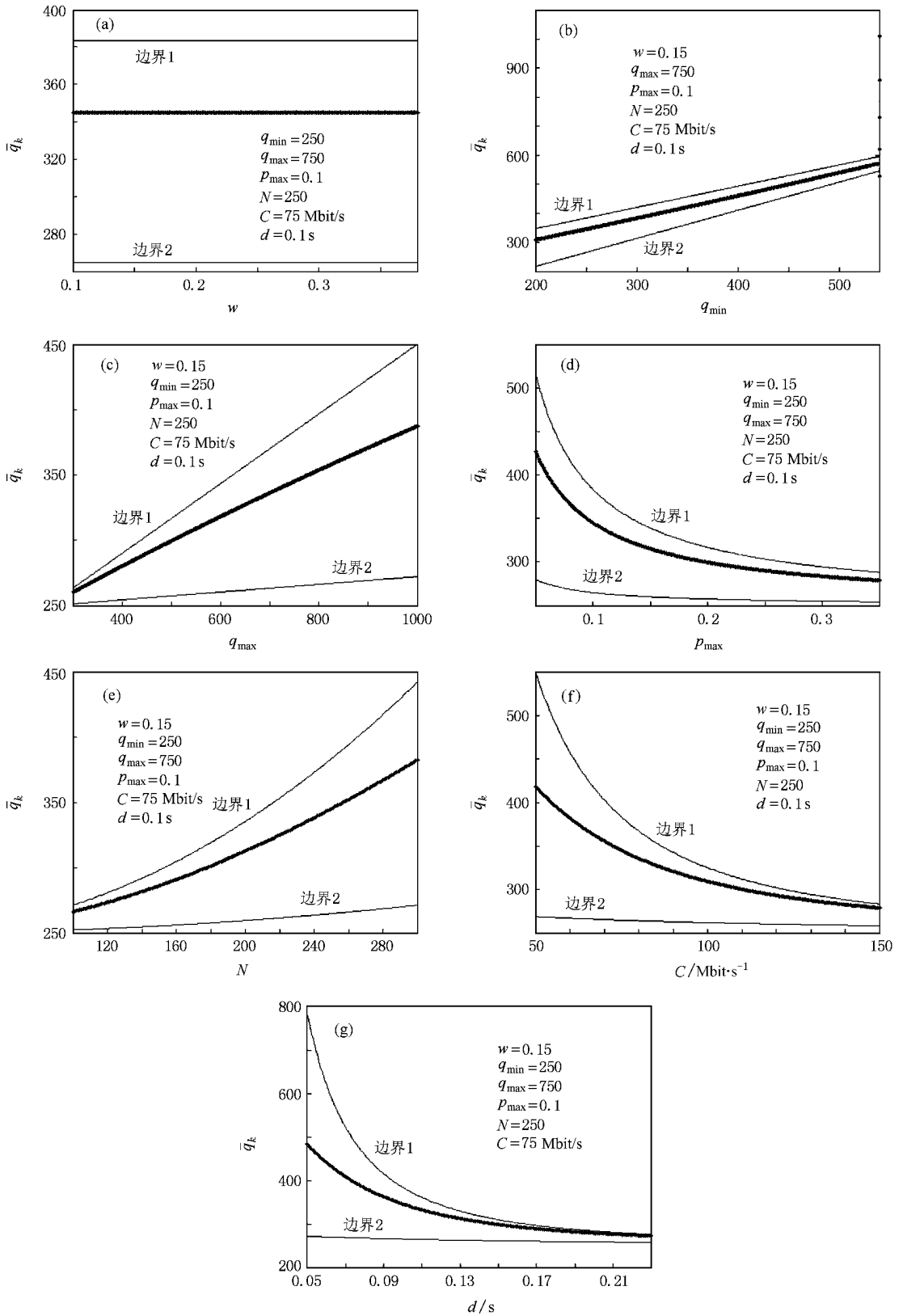


图 7 TCP-RED 系统中  $\bar{q}_k$  的控制结果 (a)  $w$  为分岔参数 ,(b)  $q_{\min}$  为分岔参数 ,(c)  $q_{\max}$  为分岔参数 ,(d)  $p_{\max}$  为分岔参数 ,(e)  $N$  为分岔参数 ,(f)  $C$  为分岔参数 ,(g)  $d$  为分岔参数

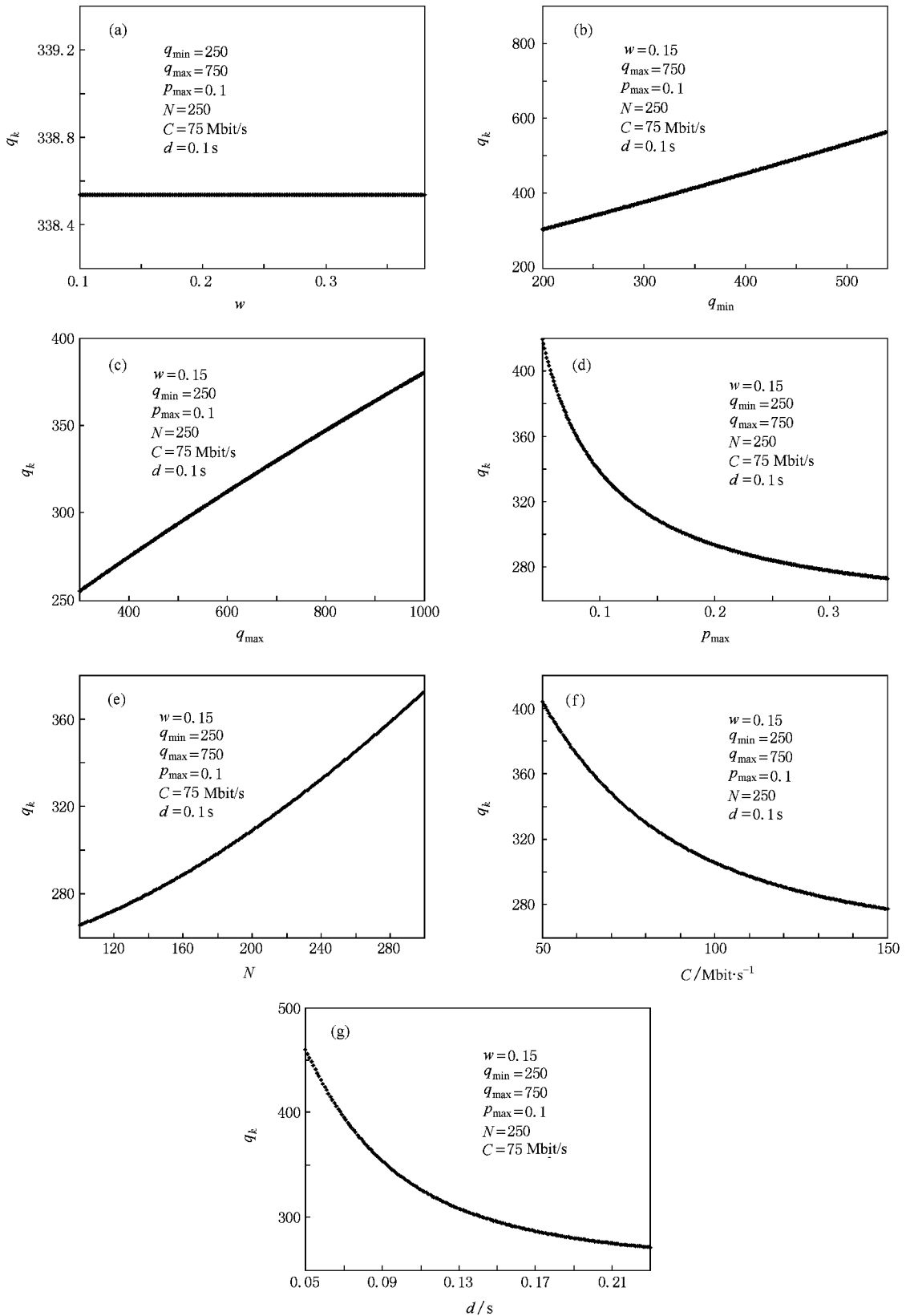


图 8 TCP-RED 系统中实施控制后  $q_k$  的状态图 (a)  $w$  为分岔参数, (b)  $q_{\min}$  为分岔参数, (c)  $q_{\max}$  为分岔参数, (d)  $P_{\max}$  为分岔参数, (e)  $N$  为分岔参数, (f)  $C$  为分岔参数, (g)  $d$  为分岔参数

宽均为 30 Mbit/s, 且队列管理采用尾部丢弃 (tail-drop) 机制. 路由器 1 和路由器 2 间的瓶颈链路带宽  $C = 150$  Mbit/s, 传播时延为 30 ms, 队列缓冲大小  $B = 7500$  包, 队列管理采用 RED 机制. 同时, RED 参数中的  $q_{\min} = 750$  包,  $q_{\max} = 2250$  包,  $p_{\max} = 1/8$ . 仿真时长为 200 s.

首先, 考虑 RED 参数中的权重系数  $w$  从小到

大变化的情况. 图 9 中给出了第 100 s 到第 200 s 内仿真得到的  $\bar{q}_k$  和  $q_k$  的值, 图中每隔 10 ms 绘制一次  $\bar{q}_k$  和  $q_k$  的值. 可以看到当  $w = 6.00 \times 10^{-6}$  时平均队列长度和瞬时队列长度的振荡情况突然明显加剧, 瞬时队列长度甚至出现了接近为零的情况, 故可以认为此时系统中发生了边界碰撞分岔, 系统迅速进入混沌状态.

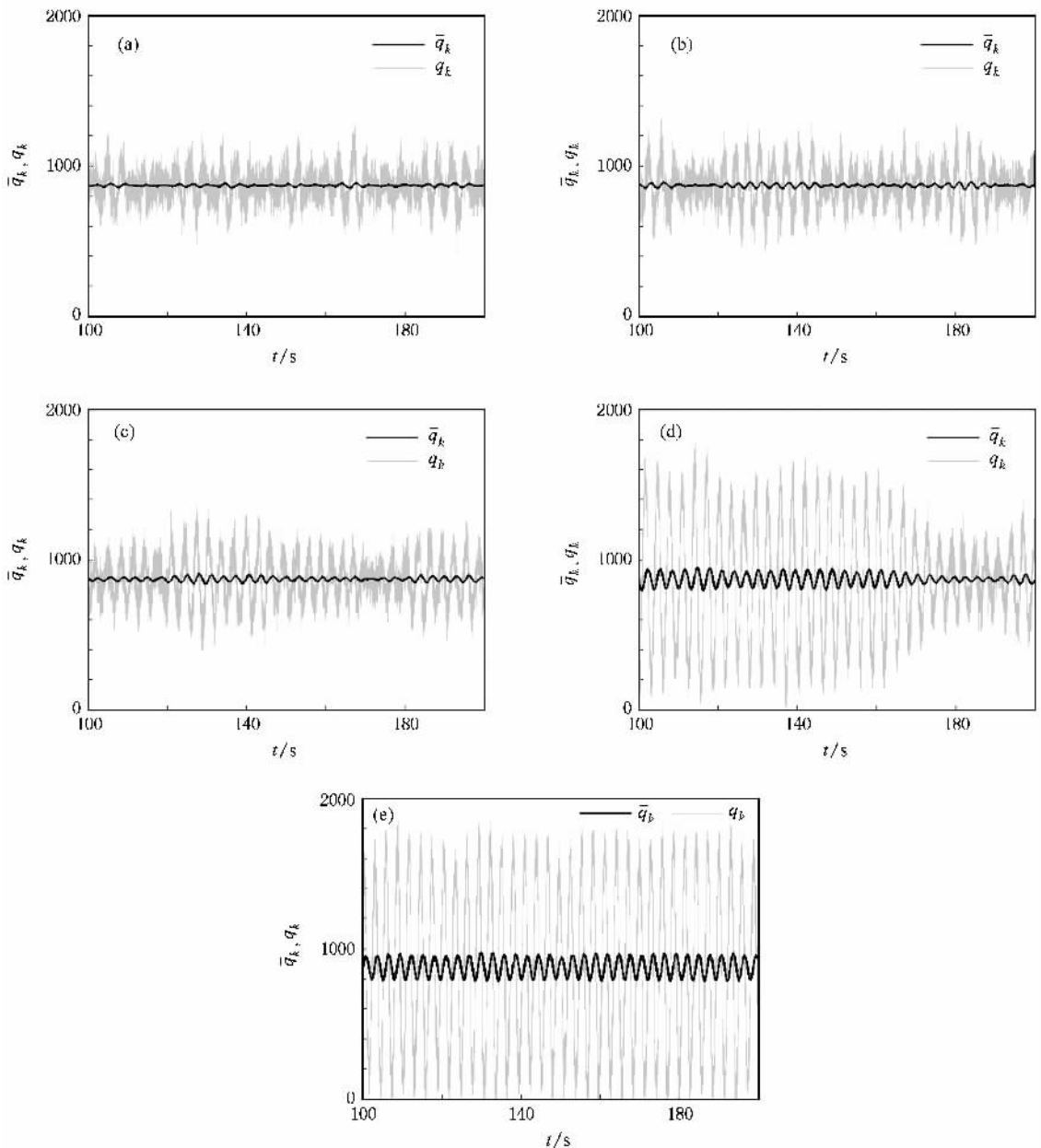


图 9 不同  $w$  值下  $\bar{q}_k$  和  $q_k$  的 NS2 仿真结果 (a)  $w = 4.00 \times 10^{-6}$ , (b)  $w = 5.20 \times 10^{-6}$ , (c)  $w = 5.90 \times 10^{-6}$ , (d)  $w = 6.00 \times 10^{-6}$ , (e)  $w = 7.00 \times 10^{-6}$

值得注意的是,在 NS2 仿真中(或实际网络中)每当有新包到达队列或队列空闲一定时间后,RED 均将更新一次  $\bar{q}_k$  和  $q_k$  的值,而由以上所述可知,在 TCP-RED 离散反馈系统模型中每隔一个网络双向时延的时长,RED 才更新一次  $\bar{q}_k$  和  $q_k$ . 根据本节中网络参数的设置可知,NS2 仿真中 RED 更新  $\bar{q}_k$  和  $q_k$  值的频率将远远高于 RED 在离散反馈系统模型中的更新频率,故 NS2 仿真中  $w$  值的设置应远

小于其在数值仿真中的设置.

由(10)–(12)式计算得到  $\bar{q}_k^*$ ,  $S$ , 并选取  $\epsilon = 50$ ,  $\gamma = -S - 0.5$ , 根据(14)和(15)式实施控制.  $\bar{q}_k$  和  $q_k$  的控制结果如图 10 所示.

然后考虑通过增加瓶颈链路的传播时延,使网络双向时延  $\mathcal{A}$  (不包括排队时延)从小到大变化的情况. 瓶颈链路的传播时延从 32 ms 逐渐增加到 40 ms, 参数  $w$  设置为  $5.20 \times 10^{-6}$ , 其他参数设置不

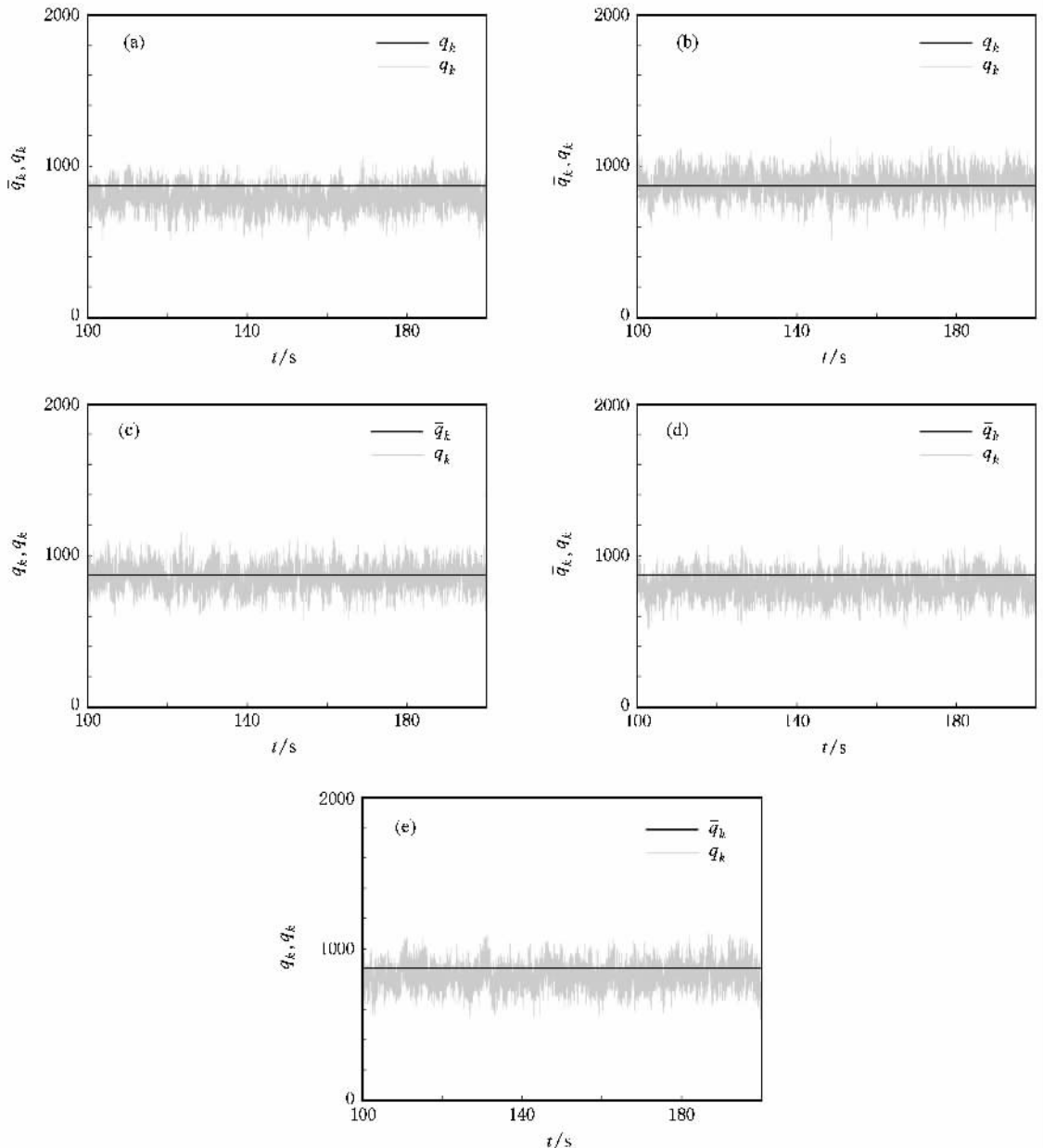


图 10 不同  $w$  值下  $\bar{q}_k$  和  $q_k$  的控制结果 (a)  $w = 4.00 \times 10^{-6}$ , (b)  $w = 5.20 \times 10^{-6}$ , (c)  $w = 5.90 \times 10^{-6}$ , (d)  $w = 6.00 \times 10^{-6}$ , (e)  $w = 7.00 \times 10^{-6}$

变. 图 11 给出了实施控制前第 100 s 到第 200 s 内  $\bar{q}_k$  和  $q_k$  的仿真结果, 图中每隔 10 ms 绘制一次  $\bar{q}_k$  和  $q_k$  的值. 可以看到当瓶颈链路的传播时延增加到 35 ms, 即网络平均双向时延(不包括排队时延)

达到 160 ms 时, 平均队列长度和瞬时队列长度的振荡情况突然明显加剧, 瞬时队列长度甚至出现了接近为零的情况, 故可以认为此时系统中发生了边界碰撞分岔, 系统迅速进入混沌状态.

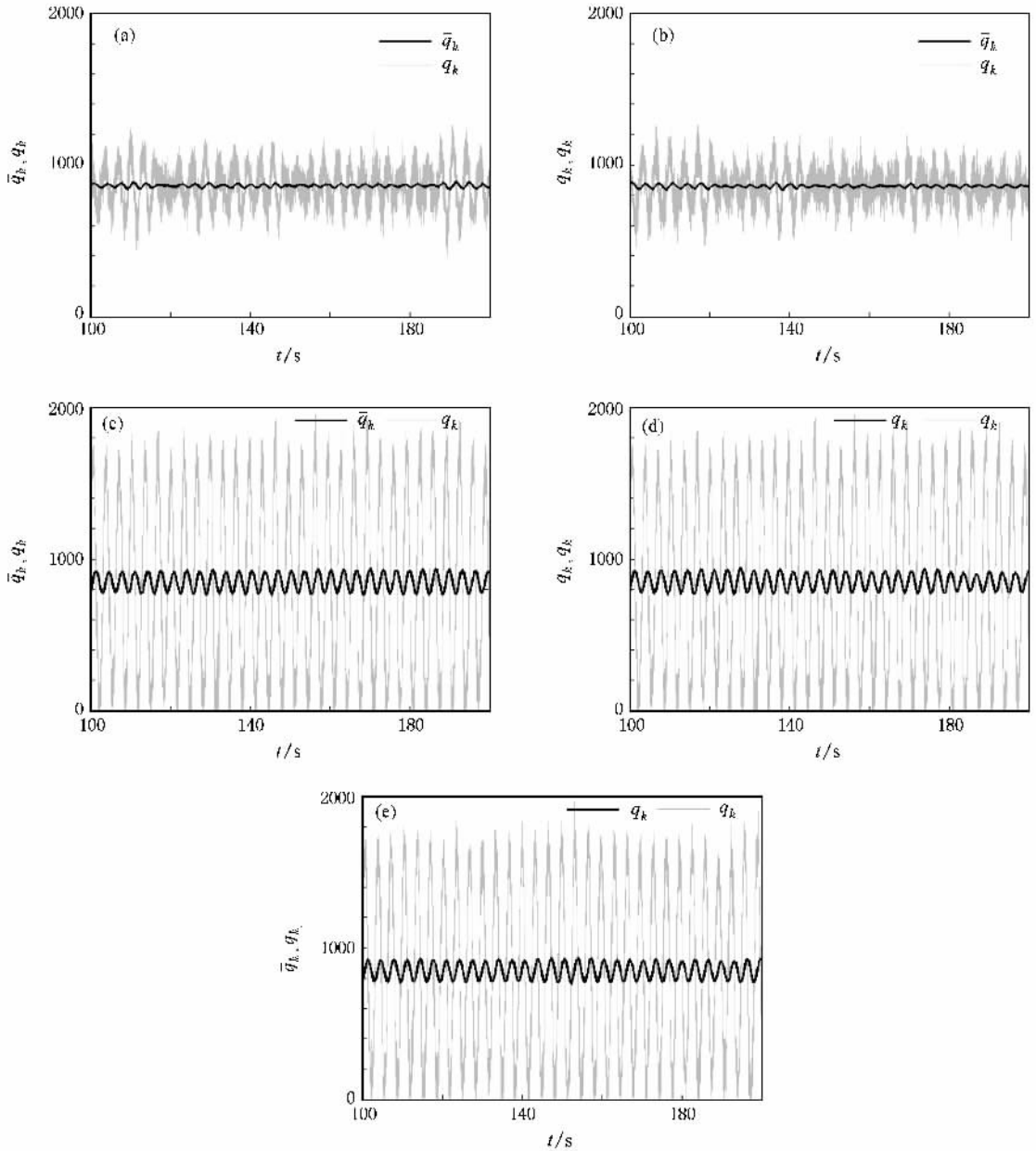


图 11 不同瓶颈链路传播时延值下  $\bar{q}_k$  和  $q_k$  的 NS2 仿真结果 (a) 瓶颈链路传播时延为 32 ms, (b) 瓶颈链路传播时延为 34 ms, (c) 瓶颈链路传播时延为 35 ms, (d) 瓶颈链路传播时延为 38 ms, (e) 瓶颈链路传播时延为 40 ms

同样由 (10)–(12) 式计算得到  $\bar{q}_k^*$ ,  $S$ , 并选取  $\epsilon = 50$ ,  $\gamma = -S - 0.5$ , 根据 (14) 和 (15) 式实施控制.  $\bar{q}_k$  和  $q_k$  的控制结果如图 12 所示.

从图 10 和图 12 可以看到, 使用该方法对系统实施控制后,  $\bar{q}_k$  和  $q_k$  均被控制在稳定的状态, 由

此系统中端到端时延抖动得到了有效遏制, 且资源利用率得到明显提高. 同时可观察到实施控制后,  $\bar{q}_k$  和  $q_k$  值所处的数值区间不会随着  $w$  的增加而变化, 但会随着  $d$  的增加而逐渐减小. 该现象与上述数值仿真的结果一致.

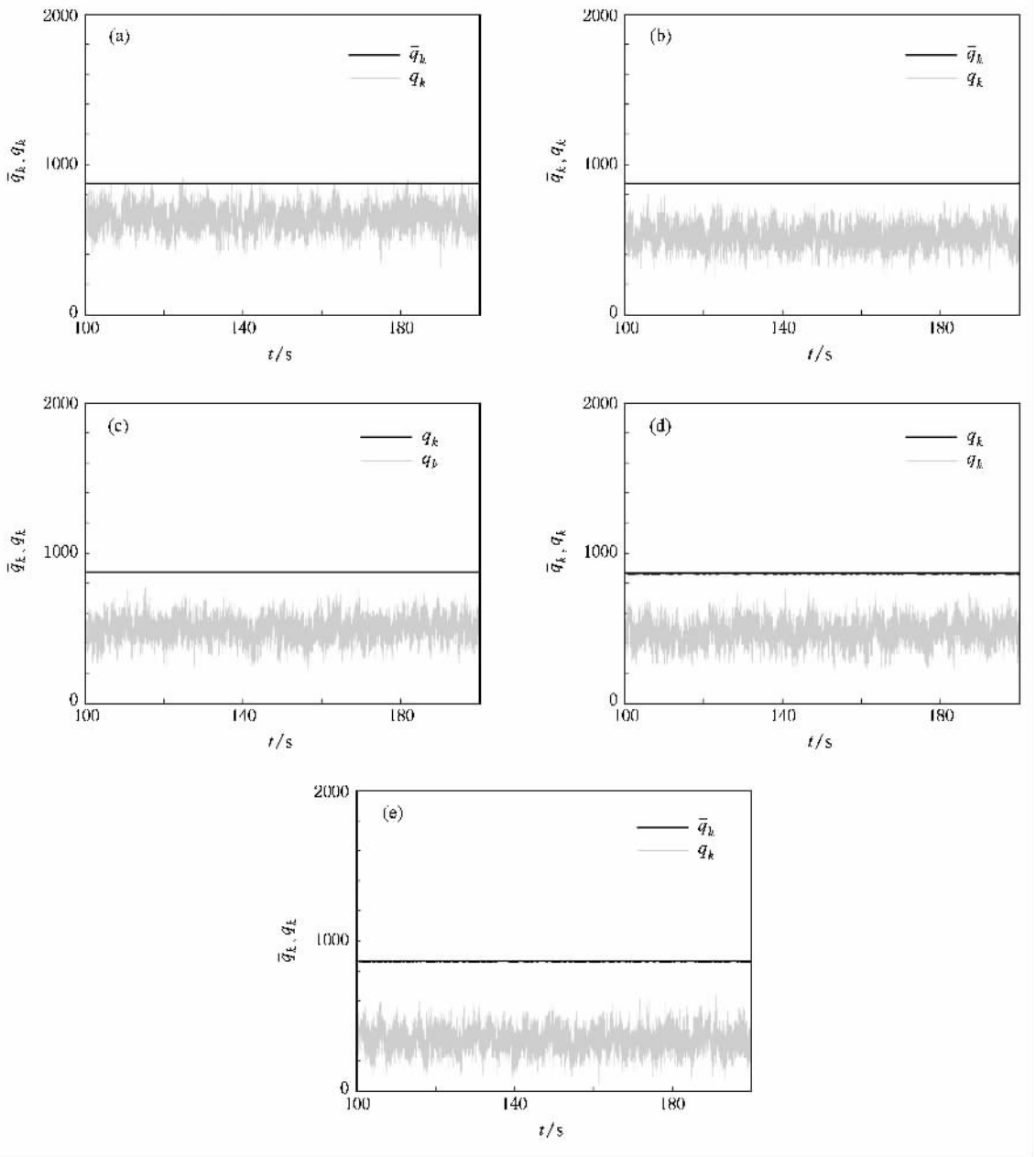


图 12 不同瓶颈链路传播时延值下  $\bar{q}_k$  和  $q_k$  的控制结果 (a) 瓶颈链路传播时延为 32 ms, (b) 瓶颈链路传播时延为 34 ms, (c) 瓶颈链路传播时延为 35 ms, (d) 瓶颈链路传播时延为 38 ms, (e) 瓶颈链路传播时延为 40 ms

## 5. 结 论

对 TCP-RED 系统中存在的边界碰撞分岔进行了深入研究. 研究表明, 当系统中的参数变化时, 系统将经历边界碰撞分岔, 在周期稳态和混沌态间

跃变. 当系统处于混沌态时, 系统中的瞬时队列长度将大幅度振荡并严重影响网络性能, 因此研究 TCP-RED 系统中的混沌控制具有重要的实用价值. 根据不动点的稳定条件提出了一种混沌控制的方法, 通过对不动点邻域内的状态变量进行扰动, 将系统稳定在不动点, 显著提高系统性能和资源利用率.

- [ 1 ] Ranjan P , Abed E H , La R J 2004 *IEEE/ACM Trans . Networking* **12** 1079
- [ 2 ] Glendinning P , Wong C H 2009 *Phys . Rev . E* **79** 5202
- [ 3 ] Banerjee S , Karthik M S , Yuan G H , Yorke J A 2000 *IEEE Trans . Circ . Syst .* **47** 389
- [ 4 ] Banerjee S , Ranjan P , Grebogi C 2000 *IEEE Trans . Circ . Syst .* **47** 633
- [ 5 ] Mathis M , Semke J , Mahdavi J , Ott T 1997 *Computer Commun . Rev .* **27** 67
- [ 6 ] Padhye J , Firoiu V , Towsley D , Kurose J 2000 *IEEE/ACM Trans . Networking* **8** 133
- [ 7 ] Hespanha J P , Bohcek S , Obraczka K , Lee J 2001 *LNCS* **2034** 291
- [ 8 ] Dai D , Ma X K , Li X F 2003 *Acta Phys . Sin .* **52** 2729 ( in Chinese ] 戴 栋、马西奎、李峰 2003 物理学报 **52** 2729 ]
- [ 9 ] Luo X S , Wang B H , Chen G R , Quan H J , Fang J Q , Zou Y L , Jiang P Q 2003 *Acta Phys . Sin .* **52** 12 ( in Chinese ] 罗晓曙、汪秉宏、陈关荣、全宏俊、方锦清、邹艳丽、蒋品群 2003 物理学报 **52** 12 ]
- [ 10 ] Li M , Ma X K , Dai D , Zhang H 2005 *Acta Phys . Sin .* **54** 1084 ( in Chinese ] 李 明、马西奎、戴 栋、张 浩 2005 物理学报 **54** 1084 ]
- [ 11 ] Liu F 2008 *Chin . Phys .* **17** 2394
- [ 12 ] Zhao X P , Schaeffer D G 2008 *Int . J . CND* **3** 041004
- [ 13 ] Chen L , Wang X F , Han Z Z 2003 *Proc . IEEE* **3** 25
- [ 14 ] Chen L , Wang X F , Han Z Z 2004 *Chaos Solitons Fract .* **21** 81
- [ 15 ] Nga J , Iu H , Ling S H , Lam H K 2008 *Chaos Solitons Fract .* **37** 977

## Border collision bifurcation and chaotic control of discrete feedback transmission control protocol-random early detection system \*

Yang Tan<sup>†</sup> Jin Yue-Hui Cheng Shi-Duan

( State Key Laboratory of Networking and Switching Technology , Beijing University of Posts and Telecommunications , Beijing 100876 , China )

( Received 17 March 2009 ; revised manuscript received 21 May 2009 )

### Abstract

Various types of border collision bifurcations occurring in a discrete feedback transmission control protocol-random early detection ( TCP-RED ) system are analyzed based on normal form method. As the system parameters change , the system experiences the border collision and a sudden jump from a periodic attractor to a chaotic attractor leading to the intense vibration of the system and serious degradation of its performance. According to the linear stability condition of fixed point , a method of chaotic control is presented. With this method , the system can be stabilized to the fixed point by perturbing the state variable in the neighborhood of the fixed point. This method can significantly improve the performance and the resource utilization of TCP-RED system , and simulation results show that it is effective.

**Keywords :** transmission control protocol , random early detection , border collision bifurcation , chaotic control

**PACC :** 0545

\* Project supported by the State Key Development Program for Basic Research of China ( Grant Nos. 2009CB320505 , 2009CB320504 ) and the National High Technology Research and Development Program of China ( Grant No. 2006AA01Z235 ).

<sup>†</sup> E-mail : atomoto@gmail.com