

# 状态无关主动队列管理算法博弈的 Nash 均衡

冯 坚, 杨路明

(中南大学 信息科学与工程学院, 湖南 长沙 410083)

**摘 要:** 状态无关的主动队列管理算法处理分组时不区分分组所在的流的信息, 因此在 Internet 中, 它易于设计和部署。文中通过数学分析和仿真方法研究 AQM 博弈和 Nash 均衡存在性。假设业务流是 Poisson 分布的且用户可自由修改发送速率, 因而有结论: Drop Tail, RED 不能获得 Nash 均衡, CHOKe 可以获得近似 Nash 均衡。依据判定条件, 推导出一种与状态无关且具有效率的 Nash 均衡 AQM 算法。

**关键词:** 拥塞控制; 主动队列管理; Nash 均衡; RED; CHOKe

**中图分类号:** TP393.01

**文献标识码:** A

**文章编号:** 1673-629X(2007)07-0127-04

## Nash Equilibria in Oblivious Active Queue Management Game

FENG Jian, YANG Lu-ming

(Information Science and Technology Academy of Central South University, Changsha 410083, China)

**Abstract:** An oblivious active queue management scheme is one which does not differentiate between packets belonging to different flows, so it is easy to implement and deploy in Internet. In this paper, study AQM game and the existence of Nash equilibria by mathematics analysis and simulation. Assume that the traffic sources are Poisson but the users can control the average rate. Find that Drop Tail and RED do not impose Nash equilibria, CHOKe can impose a nearly Nash equilibria. According to the Nash equilibria condition, a new stateless and efficient AQM scheme that impose a Nash equilibria is also presented.

**Key words:** congestion control; AQM; Nash equilibria; RED; CHOKe

## 0 引 言

在 Internet 中, 拥塞是一种持续过载的网络状态, 此时用户对网络资源的需求超过了其固有的容量。TCP 协议在确保端系统响应拥塞行为(拥塞时减小发送速率)可信性的同时, 保持了核心网络的简单性<sup>[1]</sup>, 所以 TCP 协议一直是 Internet 稳定性的保障。但视频会议、远程教学等“不响应拥塞”应用的广泛使用危害到“响应拥塞”的 TCP 应用的性能, 导致网络的拥塞瘫痪, 端系统可以随意更改速率控制方案以满足网络应用的需要, 甚至可以从软件上按自己的意图随意更改拥塞控制方案<sup>[2]</sup>, 这样, TCP 协议中端系统行为的可信度被大大降低。在当今缺乏约束的 Internet 中, 没有人积极地遵守协定, 端系统不可能心甘情愿地被束缚, 他们会采用各种方式去冲破这种束缚, 这将阻碍 Internet 的进一步发展, 所以有必要引入新的研究方法和理论以规范端系统的行为, 博弈理论为该研究提供

了坚实的数学基础<sup>[3]</sup>。

路由器是 Internet 的核心部件, 也是网络拥塞状态的最直接的感受者, 在路由器中引入相应的拥塞控制机制, 使网络本身参与资源的控制工作可以有效实现对拥塞的监测和预防。路由器中采用主动队列管理算法(AQM)来避免和管理拥塞。几年来, 对 AQM 的研究与性能评价都是从工程技术的角度出发, 如资源利用率、用户公平性等, 这些方法着重去改进和提高技术的某项或几项性能指标, 而往往忽略客观存在的非合作的端系统行为。从根本上说, 传统的拥塞控制技术, 在具备较高的技术指标的同时, 却对用户的行为缺乏合理的解释。笔者从博弈论的观点出发, 分析现行状态无关 AQM 的 Nash 均衡, 探讨一种能实现 Nash 均衡的 AQM。

AQM 分为两大类: 状态无关的和状态相关的。状态无关的 AQM, 诸如 Drop Tail 和 RED<sup>[4]</sup>处理分组时不区分分组所属的流, 即对所有流分组实施相同的操作; 状态相关的 AQM, 诸如公平队列正好相反<sup>[5]</sup>, 它对到来的分组进行判断, 以获得分组所属流的信息, 从而达到对流的管理目的。从实施的角度看, 状态相关的

收稿日期: 2006-09-23

作者简介: 冯 坚(1974-), 男, 海南琼海人, 讲师, 硕士研究生, 研究方向为计算机网络协议、企业信息化; 杨路明, 教授, 博士生导师, 研究方向为计算机体系结构、数据库及信息处理。

AQM 需要保存流的状态信息, 计算复杂, 因而很少在 Internet 中得到应用; 相反在当前的 Internet 中, 以 Drop Tail 和 RED 为代表的状态、无关的 AQM 却得到广泛应用。文中主要探讨状态无关 AQM 的 Nash 均衡。

## 1 AQM 博弈论基本模型

一个博弈的组成有: 局中人, 策略, 收益函数, 博弈结果 (Nash 均衡)<sup>[3]</sup>。

(1) 局中人 (player), 指博弈中选择行动以最大化自身利益的决策主体<sup>[3]</sup>。在 AQM 拥塞博弈中, 局中人指的是端用户, 它们为提高自己的收益可自由修改发送速率和发送参数。

(2) 策略 (strategies) 又称战略, 是指局中人应该在什么条件下选择什么样的行动, 以保证自身利益最大化<sup>[3]</sup>。论文中, 策略指用户可选择的发送速率  $\lambda_i$ 。

(3) 收益 (payoff) 又称支付, 是指局中人从博弈中获得的利益水平, 它是所有局中人策略或行为的函数<sup>[3]</sup>。用户  $i$  的收益函数规定了用户  $i$  采取行动  $\lambda_i$  时所获得的总收益, 是用户使用网络资源所实现的价值的一种度量, 它可以采用经济学中的货币度量, 也可以采用网络工程中的满意度评价指标来度量。收益函数可表达为: 收益函数 = 效用函数 - 处罚函数<sup>[3]</sup>。效用函数 (Utility Function) 规定了当没有外在的控制者时, 用户  $i$  采取行动  $\lambda_i$  时所获得的收益<sup>[3]</sup>。由于博弈中行动相关的形势, 因而用户  $i$  的效用函数不只是自身所采取行动  $x_i$  的函数, 还是其他用户所采取行动的函数。文中, 收益指用户的有效发送速率, 效用指用户的实际发送速率。

(4) 博弈的结果有两类<sup>[6]</sup>: a. 稳定的 (在相同情况下可重复的) 和可预测的 (博弈的参与者在博弈前都可预见到) 结果, 这类结果由 Nash 均衡来描述, 这是文中重点研究并希望得到的结果; b. 不稳定的、不可预测的、没有均衡的结果。

(5) Nash 均衡<sup>[3]</sup> 概念 (Nash Equilibria, NE) 是现代博弈论中的核心内容和重要基础, 许多理论研究和应用都是围绕这一基本概念展开或与之密切相关的, 文中主要讨论 AQM 博弈的 Nash 均衡。一般地, 给定  $n$  个流,  $F_1, \dots, F_n$ , 流  $F_i$  的发送速率为  $x_i$ , 有效吞吐量为  $G_i$ 。下面给出拥塞控制博弈 Nash 均衡定义。

定义 在拥塞博弈  $G = [\{X_i\}, \{G_i(\cdot)\}]$  中, 如果任一用户的发送速率  $x_i^*$ , 都是在给定其余用户行动  $x_{-i}^* = (x_1^*, \dots, x_{i-1}^*, x_{i+1}^*, \dots, x_n^*)$  的情况下, 用户  $i$  的最佳策略, 即:

$$G_i(x_i^*, x_{-i}^*) \geq G_i(x_i, x_{-i}^*), x_i \in X_i, i \in [1, n]$$

则称  $x^*$  构成  $G$  的一个 Nash 均衡。

一个行动向量构成 Nash 均衡, 意味着没有用户有单方面改变自己行动的动机, 因为这样做并不能提高自己所获得的收益。一个行动向量只有构成 Nash 均衡, 才是在现实中可能稳定存在的结果。

## 2 状态无关 AQM 的 Nash 均衡存在性判定

文中将博弈局势抽象为单瓶颈链路的情形, 如图 1 所示。在 AQM 的博弈局势中, 路由器主动队列管理算法是决定用户收益的主要因素。假定路由器中的队列模型为 M/M/1/B 队列, 路由器转发分组的速率为  $v$ 。网络中有  $n$  个业务流共享网络瓶颈, 这些流是自私的, 它们只关心自己的收益, 业务流的到达服从 Poisson 分布。流  $i$  的发送速率为  $\lambda_i$ ,  $\lambda$  为所有流总到达速率,  $G_i$  为业务流  $i$  的有效吞吐,  $p_i$  为流  $i$  经过路由器的丢包概率。

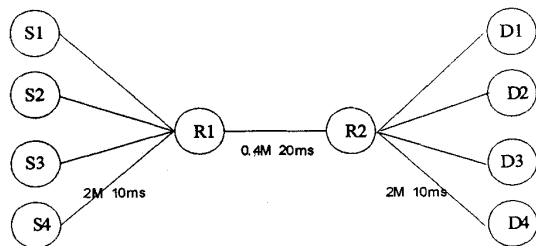


图 1 瓶颈链路

由于在 Nash 均衡时, 没有任何主体可通过增加或减少其发送速率而获得更高的收益, 因此 Nash 均衡的条件是:

$$\forall i, \frac{\partial G_i}{\partial \lambda_i} \leq 0 \quad (1)$$

为方便讨论, 在此分析对称博弈, 即所有用户具有相同的发送速率, 在 Nash 均衡点, 所有的用户具有相同的收益, 即可有以下表达式:

$$\forall i, j, G_i = G_j, \lambda_i = \lambda_j \quad (2)$$

在状态无关的 AQM 中, 丢包概率和用户的发送速率存在以下关系<sup>[7]</sup>:

$$\forall i, \frac{\partial p}{\partial \lambda_i} = \frac{dp}{d\lambda} \quad (3)$$

任一用户的收益函数可表达为:

$$G_i = \lambda_i(1 - p_i) = \lambda_i(1 - p)$$

等式两边对  $\lambda_i$  求偏导数得:

$$\frac{\partial G_i}{\partial \lambda_i} = 1 - p - \lambda_i \frac{\partial p}{\partial \lambda_i} = 0 \quad (4)$$

由于

$$\lambda_i = \frac{\lambda}{n} \quad (5)$$

由式(3)、(4)和(5),可得对称博弈的 Nash 均衡条件为:

$$\frac{dp}{1-p} = \frac{nd\lambda}{\lambda}$$

(6)

3 现行状态无关的 AQM 的 Nash 均衡存在性分析

状态无关的 AQM 算法由于实现简单和易于部署,一直是研究的热点,具有代表性的有 Drop Tail, RED 和 CHoKe<sup>[8]</sup>,本节从数学证明和仿真两方面分析它们的 Nash 均衡存在性。

3.1 Drop Tail 和 RED 队列 Nash 均衡存在性分析

定理 1: 当网络处于拥塞时,Drop Tail 不能实现 Nash 均衡。

证明:

根据排队论,缓冲容量为  $B$  的队列丢包概率为<sup>[9]</sup>:

$$p = \frac{(1-\rho)\rho^B}{1-\rho^{B+1}} \approx \rho^B$$

其中  $\rho = \frac{\lambda}{G} = \frac{\sum_{i=1}^n \lambda_i}{\sum_{i=1}^n G_i}$

(7)

网络处于拥塞时,该博弈是一常数博弈,即

$$\sum_{i=1}^n G_i = v$$

(8)

由表达式(7)和(8)知,网络拥塞时,随着  $\lambda$  的增大, $p$  增大。

$\forall \lambda_i$  给定  $\lambda_{-i}$  不变时总有  $\lambda'_i > \lambda_i$  ( $\lambda'_i$  表示业务流  $i$  在下一时刻的发送速率),使得:

$$G(\lambda'_i) > G(\lambda_i), \text{即得 } \forall i, \frac{\partial G_i}{\partial \lambda_i} > 0$$

这不符合 Nash 均衡存在的判定条件,即网络处于拥塞时,Drop Tail 不能实现 Nash 均衡,证毕。

Drop Tail 的丢包策略是分组级的,当拥塞时,其对所有的分组具有相同的丢包概率,换言之 Drop Tail 以相同的丢包概率处罚所有的流,“激进流”可通过提高发送速率,发送更多的分组,以获得更大的收益,Drop Tail 没有适合的激励机制保证用户在拥塞时降低发送速率,这意味着 Nash 均衡是不存在的。

以上定理的证明具有以下启示:主动队列管理算法的丢包策略是影响 Nash 均衡存在的关键因素;对“激进流”采取区分的更严厉的丢包惩罚是实现 Nash 均衡的有效途径。据以上分析可得,RED 对所有流的分组采取的也是“一视同仁”的丢弃概率,因此基于 RED 的主动队列管理算法中,Nash 均衡是不存在的。

通过仿真实验也验证了 Drop Tail 和 RED 不能实现 Nash 均衡。本实验采用 ns2 网络仿真工具,采用图 1 的瓶颈链路。源端至  $R_1$  的链路为 2M,10ms,  $R_2$  至目的端的链路为 2M,10ms,  $R_1$  至  $R_2$  的链路为 0.4M,20ms。

在仿真时,给定前三个流  $S_1, S_2, S_3$  的发送速率分别为:0.1、0.2、0.3 和 0.4Mb/s 时,计算流  $S_4$  分别为 0.1Mb/s~1.5Mb/s 时的有效吞吐(单位为 Mbytes),仿真时间 100s。Drop Tail 仿真数据见表 1,RED 仿真数据见表 2。

仿真结果表明,采用 Drop Tail 和 RED 时,用户端可通过提高发送速率而获得更大的收益。可见在 Drop Tail 和 RED 不能获得 Nash 均衡。

3.2 CHoKe 队列 Nash 均衡存在性分析

文献[8]提出的 CHoKe 模型中路由器不需保留任何流的状态信息,当一个分组到达路由器时,从缓冲区中随机抽取一个分组与之比较,若这 2 个分组属于同一流,则 2 个分组均被丢弃,若不属于同一流,则所到分组按一定概率  $P$  丢弃,  $P$  由当前队列长度决定(类似 RED),显然,激进流的分组被丢弃的概率高。

表 1 Drop Tail 仿真数据

$\lambda_4$ $\lambda_1\lambda_2\lambda_3$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5
0.1	1.237	2.041	2.54	2.888	3.128	3.37	3.6	3.766	3.876	3.97	4.07	4.165	4.24	4.32	4.33
0.2	0.76	1.204	1.66	1.972	2.362	2.59	2.76	2.967	3.078	3.31	3.43	3.58	3.64	3.768	3.9
0.3	0.513	0.91	1.25	1.557	1.814	2	2.25	2.491	2.624	2.77	3	3.107	3.24	3.319	3.48
0.4	0.362	0.671	0.99	1.265	1.494	1.64	1.88	2.104	2.226	2.43	2.5	2.729	2.82	2.97	3.19

表 2 RED 仿真数据

$\lambda_4$ $\lambda_1\lambda_2\lambda_3$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5
0.1	1.156	1.994	2.38	2.825	3.102	3.33	3.46	3.605	3.675	3.83	3.94	4.003	4	4.08	4.11
0.2	0.706	1.222	1.65	1.907	2.238	2.5	2.66	2.817	2.945	3.11	3.19	3.262	3.4	3.436	3.53
0.3	0.501	0.855	1.24	1.491	1.785	2.02	2.16	2.285	2.483	2.69	2.71	2.878	2.9	3.032	3.06
0.4	0.394	0.726	1	1.264	1.467	1.6	1.84	1.971	2.096	2.26	2.41	2.488	2.52	2.635	2.8

定理 2: 当网络处于拥塞时, CHOKe 不能实现严格 Nash 均衡, 但存在近似 Nash 均衡。

证明: 依据文献[8], 当一个分组到达时, 其被丢弃的概率  $p_i$  和发送速率  $\lambda_i$  存在以下关系:

$$p_i = \frac{\lambda_i}{v + 2\lambda_i}, G_i = \lambda_i(1 - 2p_i) = \frac{v\lambda_i}{v + 2\lambda_i}$$

即得

$$\frac{\partial G_i}{\partial \lambda_i} = \frac{1}{(1 + 2\frac{\lambda_i}{v})^2} > 0$$

依据 Nash 均衡存在性的判别条件, 网络中不存在严格 Nash 均衡。

因  $\frac{\partial^2 G_i}{\partial \lambda_i^2} < 0$ , 这说明随着  $\lambda_i$  增大,  $G_i$  的增长速度将减慢; 又因  $\frac{\partial G_i}{\partial \lambda_i} \ll 1$ , 这说明随着  $\lambda_i$  增大,  $G_i$  的增长幅度很小, 此时可认为网络中存在近似的 Nash 均衡。

#### 4 一种能实现 Nash 均衡的 AQM

本节讨论一种能实现 Nash 均衡的 AQM。假设  $\bar{\lambda}_n, \bar{G}_n$  和  $\bar{p}_n$  分别为  $n$  个用户在 Nash 均衡点的总发送速率、总收益和路由器丢包概率。设计的基本思路是: 依据路由器转发分组的速度  $v$ 、当前的用户数  $n$  和系统的要求给出 Nash 均衡解  $\bar{\lambda}_n$ , 由式(6)计算出 Nash 均衡点处路由器的丢包概率  $\bar{p}_n$  和总收益  $\bar{G}_n$ 。路由器将  $\bar{\lambda}_n/n$  的值作为期望发送速率通知给用户以规范用户的行为, 而路由器则以  $\bar{p}_n$  的丢包概率处理进来的分组。

设计时考虑下面三个因素:

(1) 易于部署的。为方便在 Internet 中部署, AQM 必须是尽可能状态无关的, 在本算法中需要动态统计当前用户的个数  $n$ , 依据用户的个数计算 Nash 均衡解  $\bar{\lambda}_n$  以及路由器的丢包概率  $\bar{p}_n$ 。

(2) 效率。AQM 必须具有一定的效率, 在此定义 Nash 均衡点的效率。如果满足以下条件, 就说 Nash 均衡解是有效率的<sup>[7]</sup>。

$$a. \bar{\lambda}_n(1 - \bar{p}_n) \geq c_1$$

$$b. \bar{\lambda}_n \leq c_2$$

其中  $c_1, c_2$  为常数。

(3) 可获得性<sup>[7]</sup>。当网络中用户的数量发生变化时, Nash 均衡解  $\bar{\lambda}_n$  也发生变化, 因此定义  $\omega = |\bar{\lambda}_n - \bar{\lambda}_{n-1}|$  来表示这种变化, 我们希望随着用户数量的变化,  $\omega$  的值仍然足够大<sup>[7]</sup>。

一般地给出 Nash 均衡解表达式:

$$\bar{\lambda}_n = (1 + \frac{a}{n^2 + bn + c} + \frac{d}{n + e})v \quad (9)$$

其中  $v$  为路由器转发分组的速度,  $a, b, c, d$  和  $e$  为参数, 这些参数决定了 Nash 均衡解的效率和可获得性, 下面讨论两种具体情况:

1) 假设  $n$  个用户时, Nash 均衡解为:

$$\bar{\lambda}_n = (1 - \frac{1}{4n})v$$

将上式代入式(6), 解微分方程得路由器丢包概率:

$$\bar{p}_n = 1 - \frac{1}{k\sqrt{2n-1}}$$

令  $n=1$  时,  $\bar{p}_n=0$ , 可得  $k=1$ , 即有:

$$\bar{p}_n = 1 - \frac{1}{\sqrt{2n-1}}$$

$\bar{\lambda}_n, \bar{G}_n, \bar{p}_n$  和  $n$  的关系见图 2, 可见当网络中存在较多的用户时, Nash 均衡解的效率较低的, 但  $\omega$  仍保持在一可接受的值, 即具有良好的可获得性。

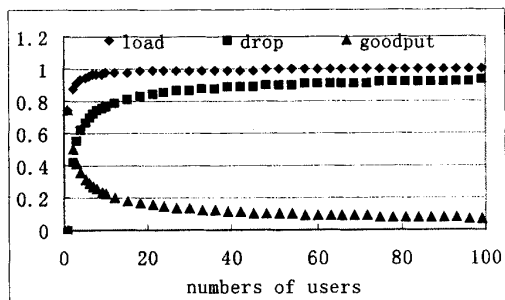


图 2  $\bar{\lambda}_n, \bar{G}_n, \bar{p}_n$  和  $n$  的关系(1)

2) 在文献[7]中, 提出的 Nash 均衡解为:

$$\bar{\lambda}_n = (1 - \frac{1}{4n^2})v$$

同理可得

$$\bar{p}_n = 1 - \frac{1}{\sqrt{3}} \sqrt{\frac{1 + \frac{1}{2n}}{1 - \frac{1}{2n}}}$$

$\bar{\lambda}_n, \bar{G}_n, \bar{p}_n$  和  $n$  的关系见图 3, 可见随着用户数目的增长, Nash 均衡解仍具有较高效率, 但  $\omega$  的值偏小, Nash 均衡点较为敏感。

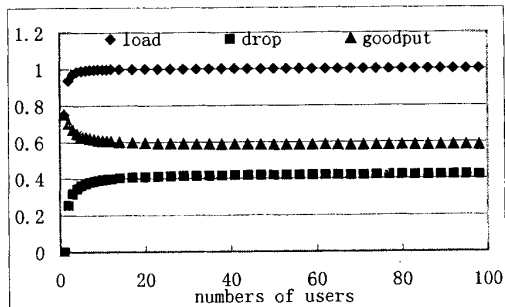


图 3  $\bar{\lambda}_n, \bar{G}_n, \bar{p}_n$  和  $n$  的关系(2)

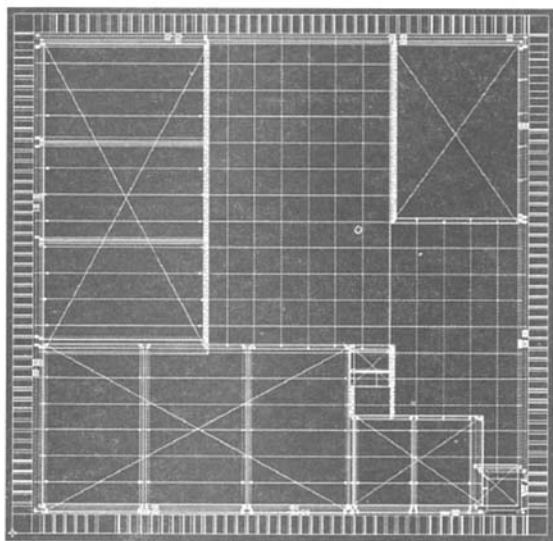


图 2 电源网络规划在实际芯片中的应用实例

\* 芯片电源环的尺寸在满足芯片本身要求的前提下,比设计规则(design rule)规定的“开槽”尺寸要小,有效降低了电子迁移现象。

\* 选择作了多条较细的电源格线,而不是少而宽的做法,实践证明这样的做法对后期的布线几乎没有产生任何不良的影响。

\* 在电源网络分析的过程中,充分利用了工具的综合和分析能力,在相对较短的时间内做了大量的尝

(上接第 130 页)

在本算法中,Nash 均衡效率和可获得性之间存在一定的矛盾,因此在实际实施时应根据系统的用户数及路由器转发分组的速度,选择适合的参数。

## 5 结束语

研究 AQM 拥塞控制 Nash 均衡的意义在于,它能有效规范端系统的不合理(自私的或贪婪的)行为,维护和实现 Internet 系统的稳定。论文提出基于 AQM 的拥塞博弈模型,并推导出 Nash 均衡的判定条件,从数学证明和仿真两方面分析了现行主动队列管理算法 Drop Tail,RED 和 CHOKe 的 Nash 均衡存在性,提出一种能实现 Nash 均衡且具有一定效率的 AQM 算法。

### 参考文献:

- [1] Altman E. Flow control using the theory of zero sum Markov games[J]. IEEE Transactions on Automatic Control, 1994, 39(4):814-818.
- [2] Friedmann E J, Shenker S. Learning and implementation on the

Internet[D]. Rutgers:Department of Economics, Rutgers University, 1998.

\* 根据最终的分析结果,获得了 8.73mV 的芯片最大压降,这对于 1.8V 供电的芯片中心区(core area)来说是极小的,可以接受的。

## 7 总结

结合 SOC 设计中的具体实践,基于复杂片上系统对于功耗完整性的特殊要求,对电源网络规划、综合和分析提出了一些建议和需要注意的问题,对提高电源规划的有效性和质量具有一定的指导意义。

### 参考文献:

- [1] Rabaey J M. Digital Integrated Circuit: A Design Perspective [M]. [s. l.]: Prentice-Hall Inc., 1996: 38-42.
- [2] Rodgers R, Technologies A, Knapp K, et al. Floorplanning Principles [C]// SNUG. San Jose: [s. n.], 2005.
- [3] Flynn J, Waldo B. Power Management in Complex SoC Design [R]. San Jose: Synopsys Support Center, 2005.
- [4] Macon G T, Roche T. Power, Analog Devices, Network Synthesis and Analysis with JupiterXT and PrimePower [C]// SNUG. Boston: [s. n.], 2005.
- [5] Synopsys. JupiterXT User Guide [M]. San Jose: [s. n.], 2004.

Internet[D]. Rutgers:Department of Economics, Rutgers University, 1998.

- [3] 黄涛. 博弈论教程 [M]. 北京: 首都经济贸易大学出版社, 2004.
- [4] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance [J]. ACM/IEEE Transactions on Networking, 1993, 1(4): 397-413.
- [5] Demers A, Keshav S, Shenker S. Analysis and simulation of a fair queuing algorithm [C]// In: Proc ACM SIGCOMM'89. Austin, Texas: [s. n.], 1989: 1-12.
- [6] 魏蛟龙. 基于博弈论的网络资源分配方法研究 [D]. 武汉: 华中科技大学, 2004.
- [7] Dutta D, Goel A, Heidemann J. Oblivious AQM and nash equilibria [J]. ACM SIGCOMM Computer Communication Review, 2002, 32(3): 20-21.
- [8] Tang Ao. Understanding CHOKe [DB/OL]. 2003-04 [2005-12-10]. Proceedings of IEEE Infocom, San Francisco, CA, 2003. <http://citeseer.ist.psu.edu/570417.html>.
- [9] McDysan D. IP 与 ATM 网络中的 Qos 和业务量管 [M]. 北京: 清华大学出版社, 2004.