

雾计算中基于古诺博弈的协作缓存优化算法

涂 亮,徐 雷

(南京理工大学 计算机科学与工程学院,江苏 南京 210094)

摘 要:随着移动互联网的快速发展,大批智能移动设备访问互联网资源,造成网络资源请求量的剧增。当大量用户终端请求网络中的流行内容时,重复传输流行内容,占用大量的带宽资源,造成整个网络拥塞加重。考虑到雾计算贴近移动终端的特点,充分利用雾服务器节点本身存储容量对用户终端请求进行存储,减少同一内容重复下载,以降低链路消耗。传统缓存算法并未考虑传输开销、时延等实际因素。对此,文中提出的基于古诺博弈的链路最小代价协作缓存算法充分考虑到上述实际因素,根据内容的缓存价值来决定内容更新,增加对较流行内容的访问。通过仿真结果可知,基于古诺博弈的链路最小代价协作缓存算法在链路消耗和缓存命中率的表现明显优于传统缓存算法。

关键词:雾计算;协作缓存;时延;链路代价;命中率

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2019)06-0013-05

doi:10.3969/j.issn.1673-629X.2019.06.003

Cooperative Cache Optimization Algorithm Based on Cournot Game in Fog Computing

TU Liang, XU Lei

(School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China)

Abstract: With the rapid development of mobile Internet, a large number of smart mobile devices access to Internet resources, resulting in a dramatic increase in network resource requests. When a large number of user terminals request the popular content in the network, the repeated transmission of popular content takes up a lot of bandwidth resources, resulting in the aggravation of the entire network congestion. Taking into account the characteristics of the fog computing close to the mobile terminal, the storage capacity of the fog server node is fully utilized to store user terminal requests, reducing the repeated downloading of the same content to reduce link consumption. Traditional caching algorithms do not consider the actual factors such as transmission overhead and delay. Therefore, we propose a link minimum cost cooperative cache algorithm based on Cournot game. Taking full account of the actual factors of both, the content update is determined according to the cache value of the content, and the access to more popular content is increased. According to simulation, it can be seen that the performance of the proposed algorithm in link consumption and cache hit rate is obviously better than that of traditional cache algorithm.

Key words: fog computing; cooperative cache; delay; link cost; hit rate

0 引言

在移动互联网快速发展的时代,大批智能移动设备访问互联网资源。显然,传统的云计算在处理如此海量的资源请求时必然会出现诸多问题,例如:高时延、移动支持性差等。雾计算可看作云计算的延伸,在云端内容中心和用户之间建立的新架构。雾计算同样可以为用户终端提供相关的计算、内容存储、请求转发等服务。由于雾计算通过将服务器节点部署距离用户

终端相对较近的地理位置,因此时延更低、对移动性支持更好^[1]。

移动互联网作为人们日常生活必不可少的部分,对音乐、视频、图片等流媒体内容的获取和发布已经成为当今互联网的主流模式。据 Cisco 流量统计数据预测:在 2014 至 2017 年间,全球的 IP 流量增长超过 4 倍。预计 2019 年,视频内容流量将达到网络流量 80%^[2]。面对流媒体内容指数式增长,移动智能设备

收稿日期:2018-06-25

修回日期:2018-10-30

网络出版时间:2019-03-06

基金项目:国家自然科学基金(61671244);中央高校基本科研业务费专项资金(30918011204)

作者简介:涂 亮(1992-),男,硕士研究生,研究方向为边缘计算;徐 雷,博士,教授,CCF 会员(57491M),研究方向为云计算、网络空间安全、SDN/NFV。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20190306.0907.028.html>

受限于其存储容量、电池容量、无线网络信号等因素,移动终端用户从云端内容中心获取的内容十分有限。那么利用雾服务器节点对经过其的请求内容进行计算并存储,可以大大降低用户终端获取该流行内容的时延、减少网络带宽资源的浪费。

目前,在已有对雾计算在移动互联网中的应用^[3-5]更多的是关注于数据在网络边缘的放置。例如,文献[6]提出一种在网络边缘节点进行计算、存储的方法;文献[7]提出对于请求相同内容的用户终端划分于同一个边缘节点,即对于单个边缘节点只存储同一种请求内容。显然,该方法不适合内容种类繁多的情况;文献[8]介绍了一种通过边缘节点自愿提供存储空间的高度集中式控制缓存方式,集中式控制方式对于地理位置分散、请求内容种类过多的情况就显得不合适。

此外,传统的缓存算法很少会对链路代价、传输时延等网络实际因素进行考虑。由于不同的请求内容情况所产生的链路开销是不一样的,所以文中提出一种基于古诺博弈的链路最小代价的协作缓存算法(link minimal cost, LMC)。该算法利用雾服务器节点对请求内容的链路代价进行实时计算,评估该请求内容对整个缓存系统的影响,以便决定是否对其进行缓存。最后对比仿真效果,该算法在命中率、链路消耗参数上表现更好。

1 雾计算中的缓存分析

本节将主要介绍雾计算中的协作缓存过程,并且对用户终端进行内容请求时产生的链路代价进行分析,同时引入节点之间协作存储的方法来提高存储效率。

1.1 缓存系统模型

雾计算中的缓存架构主要分为三部分:云端内容中心、雾计算节点集群、用户终端。现假设在仅有一个云端内容中心的情况下,雾计算服务节点集群由 $F = \{F_1, F_2, \dots, F_N\}$ 组成,其中 $F_i (i \in 1, 2, \dots, N)$ 表示第 i 个雾服务器节点。 $U = \{U_1, U_2, \dots, U_M\}$ 表示用户终端集合,其中 $U_j (j = 1, 2, \dots, M)$ 表示第 j 个用户终端。

假定在给定的地理范围,根据范围内的用户终端数 M 和雾服务器节点数 N 划分各用户终端所属雾服务器节点,即用户的请求将首先经过其所属雾服务器节点。考虑到用户终端间的移动性和社交传播影响,文中采用基于社交关系的划分方式^[9]为用户终端划分雾服务器节点,用户终端 U_i 和用户终端 U_j 的社交关系实体 $E_{i,j}$ 的计算方式如下:

$$E_{i,j} = a\lambda_{i,j} + b \frac{1}{1 + \mu D_{i,j}} \quad (1)$$

其中, μ 为比例因子; $\lambda_{i,j} \in [0, 1]$, 表示用户终端 U_i 和 U_j 的影响因子; a 和 b 分别表示内容信息和地理位置的影响因子; $D_{i,j} \in \{0, 1\}$ 表示用户终端 U_i 和 U_j 是否在影响范围内。

文中将雾服务器的缓存大小 C 划分为协作和非协作两部分(如图 1 所示)。其中非协作部分 $C(1 - R)$ 用于存放该雾服务器节点自治域内访问频率较高的内容,而协作部分 CR 通过控制协作比例 R 来缓存整个缓存系统中访问频率较高的内容。协作比例 R 的取值过小将导致整个缓存系统中的存储冗余,取值过大将导致用户终端通过其他雾服务器获取内容开销增大。文中将围绕选择一个合适的协作比例 R 来达到减少链路开销的效果。

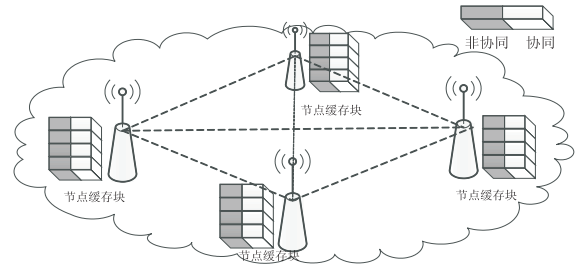


图 1 雾服务器的缓存空间组成

1.2 链路开销分析

从上一节模型结构可知,用户终端获取内容来源主要有 3 种情况:云端内容中心、其归属的雾服务器节点、其他的协作部分。文中用户终端通过不同内容源访问到内容的单位链路传输开销有所不同。单位链路传输开销为网络链路中 1 比特(Byte)数据传输需要的额外消耗。假定 $P_{i,j}$ 为雾服务器节点 F_i 和 F_j 的单位开销; P_i 为其自治域内的单位开销;雾服务器节点与云端内容中心之间的单位开销为 P_c , 其中满足 $P_c > P_{i,j} > P_i$ ^[10]。

下面分别讨论从上述 3 种情况获取内容的开销:

(1) 雾服务器节点没有存储内容 v 。用户终端将只能通过云端内容中心获取内容,则在一个缓存周期内请求该内容的总开销为:

$$P_{\text{total}} = \sum_{i \in N} T_i^v C_v (P_c + P_i) \quad (2)$$

其中, $T_i^v (1, 2, \dots, N)$ 表示内容 v 在雾服务器节点 F_i 上的访问次数; C_v 表示内容 v 的大小。

(2) 部分雾服务器节点缓存了内容 v 。用户可以直接从其归属雾服务器节点访问到内容 v 。若其归属雾服务器节点未存储内容 v , 则从距离其最近的雾服务器节点获取。那么获取该内容的总开销为:

$$P_{\text{total}} = \text{save}(N^v) P_s C_v + \sum_{i \in N \setminus \{N^v\}} T_i^v C_v (\min(P_{i,j} + P_i)) \quad (3)$$

其中, $N \setminus \{N^v\}$ 表示没有存储内容 v 的雾服务器

节点的集合; $\text{save}(N^v)$ 为存储了内容 v 的雾服务器节点个数; P_s 为雾服务器节点的单位数据的存储开销; $\min(P_{i,j})$ 为雾服务器节点之间的最小链路开销。

(3) 各雾服务器节点均存储了内容 v 。用户可以直接在其归属雾服务器节点访问到内容 v , 那么获取该内容总开销为:

$$P_{\text{total}} = \text{save}(N) P_s C_v \quad (4)$$

其中, $\text{save}(N)$ 为整个缓存系统雾服务器节点个数。

1.3 缓存价值计算

对上一节的链路开销分析, 本节对获取雾服务器节点 F_i 上的内容 v 所减少的链路开销称为缓存该内容的缓存价值 value_i^v 。根据内容 v 在整个缓存系统中的存储情况, 其缓存价值 value_i^v 的具体表达式如下:

(1) 各雾服务器节点没有存储内容 v , 则缓存价值 value_i^v 为:

$$\text{value}_i^v = T_i^w P_i + \sum_{j \in N \setminus \{i\}} T_j^w (P_c + P_i - P_{i,j}) - P_s C_v \quad (5)$$

其中, $T_i^w P_i$ 表示雾服务器节点 F_i 其自治域内的用户终端直接获取内容 v 而节省的开销; $\sum_{j \in N \setminus \{i\}} T_j^w (P_c + P_i - P_{i,j})$ 表示雾服务器节点 F_i 其自治域的用户终端从其他协作雾服务器节点 F_j 获取内容 v 而节省的开销; $P_s C_v$ 表示雾服务器节点 F_i 存储该内容 v 的存储开销。

他协作雾服务器节点 F_j 获取内容 v 而节省的开销; $P_s C_v$ 表示雾服务器节点 F_i 存储该内容 v 的存储开销。

(2) 部分雾服务器存储了内容 v , 那么缓存价值 value_i^v 为:

$$\text{value}_i^v = T_i^w \min(P_{i,j}) + \sum_{k \in N \setminus \{i\}} T_k^w [\min(P_{k,j}) - \min(P_{i,j}) + P_i] - P_s C_v \quad (6)$$

其中, $T_i^w \min(P_{i,j})$ 为雾服务器节点 F_i 其自治域内的终端直接访问到内容 v 而不从其他雾服务器节点 F_j 访问内容所减少的开销; 后半部分为其他雾服务器节点 F_k 就近从雾服务器节点 F_i 访问到内容 v 而节省的开销; $P_s C_v$ 表示 F_i 存储内容 v 的存储开销。

2 链路最小代价缓存算法

前一节详细介绍了雾计算中协作缓存的组成部分及缓存价值的计算。本节将主要介绍基于古诺博弈的链路最小代价缓存算法的实现, 该算法由两部分组成: 缓存决策部分和缓存替换部分^[11-12]。

2.1 缓存决策算法

决策流程如图2所示。各个雾服务器节点通过协作部分存储系统中较为流行的内容, 云端为整个雾计算提供请求内容、计算、协作调度, 其中关于古诺博弈的协作调度实现将在下一节介绍。

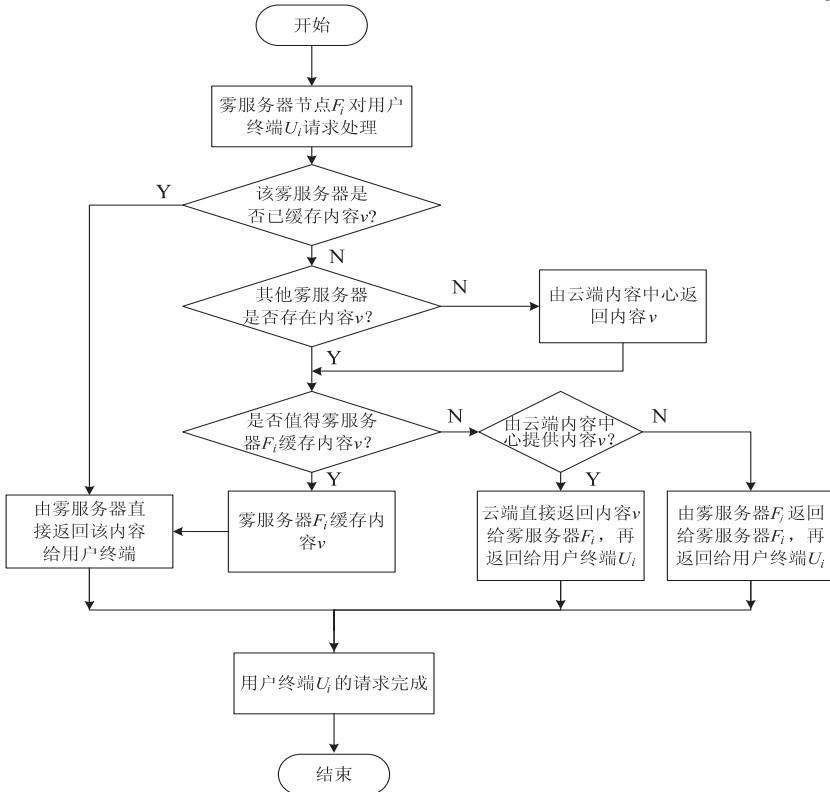


图2 缓存决策流程

2.2 缓存替换算法

在用户终端 U_i 所需的内容 v 在其所属的雾服务器节点 F_i 中没有存储时, 雾服务器节点 F_i 依据内容 v

缓存价值决定存储内容、如何更换其缓存空间中已存储的内容, 其中文中假定缓存块大小和内容块大小相同(下同)。

假设用户终端 U_i 请求的内容 v 返回至雾服务器节点 F_i , 若雾服务器节点 F_i 非协作部分有空余缓存空间, 则直接存储内容 v 。若非协作部分已经没有缓存空间, 则继续查找协作部分是否还有缓存空间。若协作部分还有空间, 则将非协作部分缓存价值最大的内容移置协作部分, 同时更新云端中关于协作部分的内容信息, 之后将非协作部分中缓存价值最大的内容更换为内容 v 。

否则对比内容 v 和非协作部分缓存价值最小的内容, 若缓存价值大于最小内容, 则执行替换操作, 否则不执行替换。

2.3 古诺博弈的协作调度

在 1.1 节中描述了各雾服务器节点的协作部分组成缓存空间, 可以为该协作缓存空间建立一个市场垄断模型。即当不同雾服务器节点均未存储所需的同一个内容时, 那么可以将该请求内容看作商品。为了保证整个缓存系统中该内容不重复存储, 各雾服务器节点均希望能就近存储其缓存空间中, 那么节点之间便通过竞争以获得最大的收益。最后由云端内容中心将该内容下发到雾服务器节点, 这样可以使得整个缓存系统的链路开销达到最小, 这就是典型的古诺博弈。

运用古诺博弈, 设定每个雾服务器节点均为出价者。每个出价者给出其对该请求内容的出价 p_i , 云端内容中心作为唯一的内容源提供者, 对各雾服务器的出价进行权衡。因此, 得到相应的定价函数^[13]如下:

$$c(P) = a + b \left(\sum_{i=1}^N p_i \right)^{\tau} \tag{7}$$

其中, a, b 和 τ 均表示正常数; $P = \{p_1, p_2, \dots, p_N\}$ 表示不同雾服务器节点的出价集合。令 $\tau \geq 1$, 即表明该定价函数为凸函数, 从而确保了云端内容中心将相同的请求内容下发到使整个缓存系统链路开销最小的雾服务器节点上。

从 1.2 节链路开销分析可得各个雾服务器节点节省的链路开销为:

$$L^{\text{save}} = L^{\text{nocache}} - L^{\text{cache}} = n_i \cdot \frac{n_i^{\text{cache}}}{n_i} \cdot P_c \cdot C_i - c(P) \cdot d_i \tag{8}$$

其中, $\frac{n_i^{\text{cache}}}{n_i}$ 为雾服务器节点 F_i 中已存储的内容占其自治域内请求内容总和的比例; P_c 为云端内容中心到用户终端的单位传输开销, 即 $P_c = P_i + P_e$; $c(P)$ 为单位缓存开销; $d_i = N_i \cdot C_i$ 为请求内容大小总和。

根据文献[14]对式 8 中 $\frac{n_i^{\text{cache}}}{n_i}$ 转换为 $\sum_{i=1}^{N_i} i^{-\sigma} \approx \frac{N_i^{1-\sigma}}{1-\sigma}$, 其中 σ 为内容流行度分布因子, 可以得到各雾

服务器节点基于古诺博弈^[15]的缓存模型的效用函数:

$$\pi_i(P) = n_i \cdot \frac{N_i^{1-\sigma}}{(1-\sigma) \omega_k} \cdot P_c \cdot C_i - c(P) \cdot d_i \tag{9}$$

其中, ω_k 表示请求内容的概率总和。

3 仿真分析

为了验证提出的基于古诺博弈的链路最小代价算法 (LMC) 的有效性, 选择了传统缓存算法: 最近最小使用 (least recency user, LRU)、先来先服务 (first in first out, FIFO)、随机 (random) 算法作为对比算法。假定所有用户终端所有请求均满足 Zipf 分布^[13]。同时, 为了评价算法的性能, 定义了三个参数作为参照: 命中率、缓存时延、缓存链路开销。其中仿真参数如表 1 所示。

表 1 仿真参数具体设置

参数	设置
地理范围 D/km^2	1
请求内容数 v	2 500
雾服务器数 F	[2, 8]
用户终端数 M	1 000
Zipf 参数 σ	[0.6, 1.2]
协同比例 R	[0.4, 0.8]
缓存周期 T/s	3 600
缓存空间 C	[100, 200]
古诺参数 a, b, τ	0, 1, 1

从图 3 可知, 各算法平均缓存命中率在 σ 增加时均有所增加。这是因为当内容流行度增大, 用户终端对流行内容的请求概率增大, 雾服务器节点对请求内容的替换频率降低, 命中率增加。文中提出的 LMC 算法相比传统算法, 通过雾服务器之间的协作缓存, 使得热点内容停留在缓存中的时间更长, 缓存命中率也随之增加。

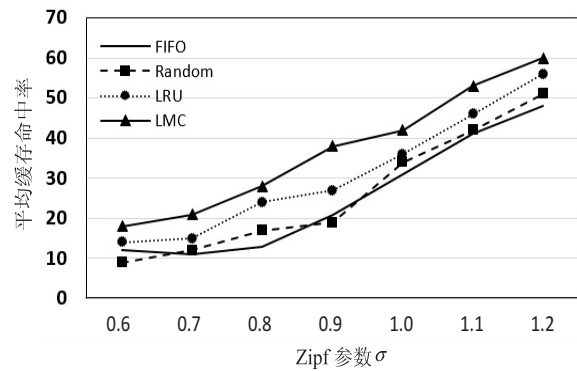


图 3 缓存命中率对比

从图 4、5 可知, 各算法的平均缓存时延和链路开

销随着 Zipf 参数 σ 的增加均减少。因为参数 σ 的增加,用户终端请求同一内容的概率增大,相应的内容可以直接从本地雾服务器节点获取,减少了从云端内容中心获取内容而产生的传输时延和链路开销。文中提出的 LMC 算法相比传统算法,通过协作缓存,丰富了整个缓存系统的内容种类,从而比传统算法拥有更低的时延和更小的链路开销。

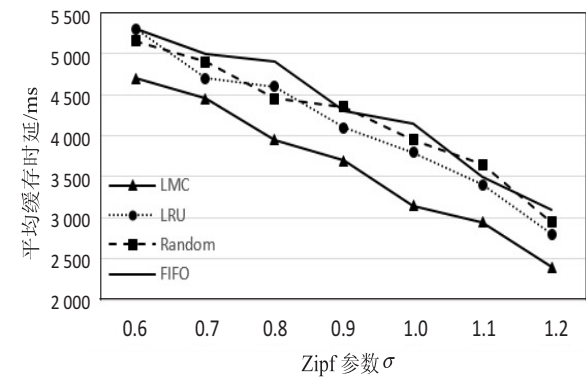


图4 缓存时延对比

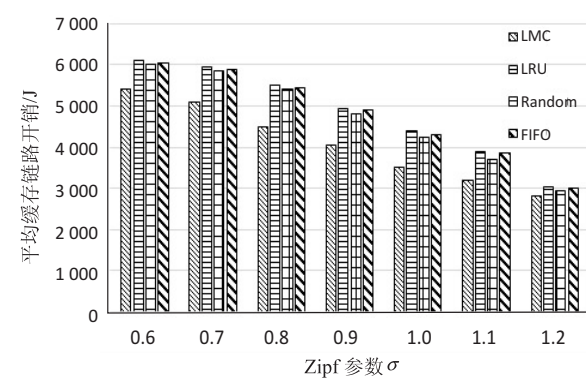


图5 缓存链路开销对比

同理,该算法对于协作比例 R 、雾服务器节点个数 F 的参数影响均优于传统算法的仿真结果,由于篇幅有限,不作论述。

4 结束语

文中首先对雾计算中的缓存系统架构进行介绍,同时提出协作缓存对缓存优化的必要性。提出的基于古诺博弈的链路最小代价算法充分考虑了不同内容的流行度、地理位置、各自自治域的链路代价等因素,各协作部分通过古诺博弈,充分利用协作缓存空间,从而减少缓存系统中缓存相同的内容,增加了内容丰富度。仿真效果表明,LMC 算法在缓存命中率、链路时延、链路开销的表现更好。

参考文献:

[1] 杨志和. 物联网的边界计算模型:雾计算[J]. 物联网技术, 2014(12):65-67.

[2] 吴磊. 思科:“雾计算”的 buzz 与干货[J]. IT 经理世界, 2014(11):68-69.

[3] 施巍松,孙辉,曹杰,等. 边缘计算:万物互联时代新型计算模型[J]. 计算机研究与发展,2017,54(5):907-924.

[4] 田辉,范绍帅,吕昕晨,等. 面向 5G 需求的移动边缘计算[J]. 北京邮电大学学报,2017,40(2):1-10.

[5] 戴晶,陈丹,范斌. 移动边缘计算促进 5G 发展的分析[J]. 邮电设计技术,2016(7):4-8.

[6] PENG Mugen, YAN Shi, ZHANG Kecheng, et al. Fog-computing-based radio access networks: issues and challenges[J]. IEEE Network, 2015, 30(4):46-53.

[7] TAO Meixia, CHEN Erkai, ZHOU Hao, et al. Content-centric sparse multicast beamforming for cache-enabled cloud RAN[J]. IEEE Transactions on Wireless Communications, 2016, 15(9):6118-6131.

[8] CHANDRA A, WEISSMAN J, HEINTZ B. Decentralized edge clouds[J]. IEEE Internet Computing, 2013, 17(5):70-73.

[9] 邢起源,王菁,闫阿宾,等. 一种基于社交关系的移动缓存替换算法[J]. 计算机科学,2016,43(6):44-49.

[10] 刘银龙,汪敏,马伟,等. P2P 缓存系统中总开销最小的协作缓存策略[J]. 通信学报,2015,36(3):2015073-1-2015073-8.

[11] SARAYDAR C U, MANDAYAM N B, GOODMAN D. Efficient power control via pricing in wireless data networks[J]. IEEE Transactions on Communications, 2002, 50(2):291-303.

[12] BRESLAU L, CAO Pei, FAN Li, et al. Web caching and Zipf-like distributions: evidence and implications[C]//Eighteenth joint conference of the IEEE computer and communications societies. New York, NY, USA: IEEE, 2002:126-134.

[13] NIYATO D, HOSSAIN E. A game-theoretic approach to competitive spectrum sharing in cognitive radio networks[C]//IEEE wireless communications and networking conference. Kowloon, China: IEEE, 2007:16-20.

[14] 张国鹏. 基于博弈论的无线网络资源竞争与协作机制研究[D]. 西安:西安电子科技大学,2009.

[15] 林昌雷. 基于博弈论的云计算资源分配机制研究[D]. 杭州:杭州电子科技大学,2014.