

## 基于非协作博弈的边缘分布式缓存方案\*

顾汇贤<sup>1</sup>, 王海江<sup>1</sup>, 魏贵义<sup>2</sup>

<sup>1</sup>(浙江科技学院 信息与电子工程学院, 浙江 杭州 310023)

<sup>2</sup>(浙江工商大学 计算机与信息工程学院, 浙江 杭州 310018)

通信作者: 王海江, E-mail: wanghaijiangyes@163.com; 魏贵义, Email: weigy@mail.zjust.edu.cn



**摘要:** 随着多媒体数据流量的急剧增长, 传统云计算模式难以满足用户对于低延时和高带宽的需求. 虽然边缘计算中基站等边缘设备拥有的计算能力以及基站与用户之间的短距离通信能够使用户获得更高的服务质量, 但是如何利用边缘节点的收益和成本之间的关系设计边缘缓存策略, 仍然是一个具有挑战性的问题. 利用 5G 和协作边缘计算技术, 在大量短视频应用场景下, 提出了一种协作边缘缓存技术来同时解决以下 3 个问题: (1) 通过减少传输延时, 提高了用户的服务体验; (2) 通过近距离传输, 降低了骨干网络的数据传输压力; (3) 分布式的工作模式减少了云服务器的工作负载. 首先定义了一个协作边缘缓存模型, 其中, 边缘节点配备有容量有限的存储空间, 移动用户可以接入这些边缘节点, 一个边缘节点可以服务多个用户; 其次, 设计了一个非协作博弈模型来研究边缘节点之间的协作行为, 每一个边缘节点看成一个玩家并且可以做出缓存初始和缓存重放策略; 最后, 找到了该博弈的纳什均衡, 并设计了一个分布式的算法以达到均衡. 实验仿真结果表明, 提出的边缘缓存策略能够降低用户 20% 的延时, 并且减少了 80% 的骨干网络的流量.

**关键词:** 非协作博弈; 纳什均衡; 分布式缓存放置; 缓存命中率; 平均边缘效用

**中图法分类号:** TP391

中文引用格式: 顾汇贤, 王海江, 魏贵义. 基于非协作博弈的边缘分布式缓存方案. 软件学报, 2022, 33(11): 4396-4409. <http://www.jos.org.cn/1000-9825/6322.htm>

英文引用格式: Gu HX, Wang HJ, Wei GY. Distributed Edge Caching Scheme Using Non-cooperative Game. Ruan Jian Xue Bao/ Journal of Software, 2022, 33(11): 4396-4409 (in Chinese). <http://www.jos.org.cn/1000-9825/6322.htm>

### Distributed Edge Caching Scheme Using Non-cooperative Game

GU Hui-Xian<sup>1</sup>, WANG Hai-Jiang<sup>1</sup>, WEI Gui-Yi<sup>2</sup>

<sup>1</sup>(School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China)

<sup>2</sup>(School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China)

**Abstract:** Due to the rapid growth of multimedia data traffic, the traditional cloud computing model has been greatly challenged in satisfying users' demands for low latency and high bandwidth. Therefore, edge computing is becoming an emerging computing paradigm. The computing capacity of edge devices such as base stations and the short distance between users and base stations enable users to obtain higher service quality. It is still a challenging problem to design edge caching strategy based on the relationship between benefits and costs of edge nodes. Using 5G and collaborative edge computing technology, in a large number of short video application scenarios, this study proposes a collaborative edge caching technology to simultaneously solve the following three problems: (1) by reducing the transmission delay, to improve users' service experience; (2) by cutting down transmission latency to reduce the data transmission pressure of the backbone network; (3) through distributed computing to reduce the workload of the cloud servers. First, a collaborative edge caching model is defined where the edge nodes are equipped with limited storage space, mobile users can access to edge nodes, one node can serve multiple users. Second, a non-cooperative game model is designed to study the cooperative behavior between edge nodes.

\* 基金项目: 国家自然科学基金(U1709217); 浙江省自然科学基金(LQ20F020010)

收稿时间: 2020-05-30; 修改时间: 2020-12-30; 采用时间: 2021-02-22

Each edge node is treated as a player and can make cache initialization and cache replacement strategies. Thirdly, the Nash equilibrium of the game is found, and then a distributed algorithm is designed to reach the equilibrium. Finally, the simulation results show that the proposed edge caching strategy can reduce the latency of users by 20% and reduce the traffic of backbone network by 80%.

**Key words:** non-cooperative game theory; Nash equilibrium; distributed cache placement; cache hit rate; average edge utility

随着 5G 网络的广泛应用, 边缘计算正在成为一种重要的计算模式, 该模式可以在网络边缘为用户提供高带宽的互联网服务. 同时, 随着笔记本、手机、智能穿戴设备等新型智能终端的计算存储能力的不断提升, 终端用户在边缘网络和互联网之间生成、检索和共享大规模数据和文件的情况越来越普遍. 在线/移动社交媒体和网络的快速增长进一步加速了这个趋势: 视频片段越来越多地由用户生成并立即与他们的朋友分享. 因此, 移动用户在访问云服务, 尤其是边缘的多媒体服务时, 对高质量的体验有着很高的期望. 据预测: 到 2020 年, 全球移动数据流量将达到 30.6 EB/月<sup>[1]</sup>. 这种爆炸性的流量已经对现有的网络架构造成了很大的负担<sup>[2]</sup>, 同时也应该采用新的技术提高目前蜂窝网络的容量<sup>[3]</sup>.

大量的研究表明, 互联网上的很大一部分流量负荷是由于热门文档的反复浏览和下载造成的<sup>[4-5]</sup>. 在传统的云计算模型中, 数据处理设备和复杂软件位于多个数据中心或云中心<sup>[6]</sup>. 由于用户的需求和云服务器的供应位于互联网的不同边缘, 用户必须通过整个互联网访问数据中心. 需求和供应之间的长距离传输导致了极高的延时, 从而导致用户无法获得高质量的网络服务. 尤其是来自流行短视频应用比如斗鱼、抖音等, 给骨干网络造成了巨大的压力. 因此, 为了提升用户体验, 需要利用移动终端和云服务器的优化技术来提高流量密集型移动应用的服务质量.

众所周知, 内容的重复传输会导致大多数边缘到边缘的数据流量. 一些研究人员尝试使用 IP 组播技术<sup>[7,8]</sup>来降低主干网络的压力. IP 组播技术作为减少冗余流量问题的在线解决方案, 允许应用程序在协议级解决问题, 而不会对边缘网络产生重大影响. 但由于用户的移动性, 边缘网络的拓扑结构可能发生快速变化, 在移动边缘网络中实现组播协议例程(如 TTL 递增、遍历路由器和 IGMP 的加入)也并不容易. 事实上, 在异构移动边缘网络中, 应用程序和协议的性能可扩展性, 是 IP 组播技术在处理重复数据流量减少问题时所面临的主要挑战.

另一种解决方案是离线技术, 比如提前获取和将数据保存在几个特定的服务器上(被称为内容分发网络, CDN 服务器)<sup>[9,10]</sup>, 这种用来存放用户数据的服务器被称为 CDN 服务器. CDN 服务器比云服务器距离用户更近. 离线技术充分利用了网络的内在时空特性, 网络在 CDN 服务器相对空闲的时候预取和保留数据, 响应后来的用户需求. 为了给用户提供有竞争力的服务质量, 大多数应用程序将内容交付外包给商业 CDN, 允许将流行的内容缓存在中间 CDN 服务器中. 因此, 用户可以通过中间 CDN 访问流行内容而不需要从远程服务器<sup>[11]</sup>进行重复传输. 例如, 为了减少核心网络上的冗余传输, 提出了在演进的包核(EPC)和基站(BS)上进行主动缓存的方法<sup>[12,12]</sup>. 然而, 一方面, 用户在 EPC 上得到的内容也会导致产生大量的重复流量成本<sup>[11]</sup>; 另一方面, CDN 的部署是非常昂贵的, 互联网服务提供商(ISP)不能将所有的内容发布到 CDN, 尽管使用了互联网的运营商 CDN<sup>[13]</sup>.

P2P 网络也被提出来解决上述问题<sup>[14-16]</sup>. 它有利于在大量分布式用户之间共享大量文件. 但是由于应用程序不容易通过 P2P 网络管理其业务流程, 因此适合在线短视频服务. 目前, 编码缓存方案(coded cache scheme, CCS)正成为这一领域的热点<sup>[7]</sup>. CCS 技术旨在通过对多用户需求的数据进行编码, 减少数据传输的重复和骨干网络的带宽消耗. CCS 利用用户需求的总体全局缓存大小和多样性来生成用于骨干网络传输的编码数据. 从信息论的角度来看, CCS 方案动态地减少了非合作用户之间的数据传输. 然而, CCS 最大的挑战是解决用户缓存大小、所需数据内容的普及程度和边缘网络拓扑的异构性. 到目前为止, CCS 还只是一种理论方法, 在移动边缘网络中并不实用.

为了解缓主干网和云服务器的流量压力, 提出了移动边缘缓存<sup>[17]</sup>. 通过边缘缓存, 预取的数据内容存储在边缘设备上, 如基站、终端设备或专门设计的边缘节点上. 对于单一边缘节点, 边缘缓存的性能受限于有限的缓存大小<sup>[18,19]</sup>. 提高缓存命中率的主要解决方案是通过研究数据内容的受欢迎程度和用户的偏好<sup>[20]</sup>来设计更有效的内容放置算法. 在一些合作机制下, 网络化的边缘节点可以发展成为一个巨大的边缘缓存池, 为移

动终端用户提供数据访问服务<sup>[21]</sup>。与远程服务器相比,边缘节点通过更短的路径(有时是一跳无线传输)向用户交付预取的数据内容,而不会导致在主干链路或核心网络上的数据传输<sup>[2,22-25]</sup>。

在 5G 网络中,网络切片技术可以将一个网络切割成多种边缘网络<sup>[26,27]</sup>。5G 基础设施可以被分成几个部分,专门用于不同的定制网络服务。在这种应用程序范例中,分割的资源被分配给每个应用程序。我们知道,边缘网络的主要资源是无线通信带宽、计算能力和边缘服务器的存储能力。切片 5G 网络为边缘用户接入互联网提供了灵活的网络资源。但是由于切片边缘服务器的存储资源有限,在一个边缘服务器上进行缓存或者在多个边缘服务器之间进行协同缓存,对于分组低延时、高带宽短视频文件访问的应用场景来说容量是不够的。我们将前文中提及的离线技术、P2P 网络以及边缘缓存各自的使用场景及优劣整理成表 1。

表 1 缓解网络流量压力的主要解决方案的对比

技术	使用场景	优势	劣势
离线缓存	视频/音频点播 视频直播加速	减少带宽的使用, 实现跨区域的全覆盖	从 CDN 到终端用户之间仍然存在大量的 内容冗余,并且 CDN 的部署价格昂贵
P2P 网络	在线短视频服务 流媒体直播	可以向其他用户提供服务,也可以 从其他节点得到服务,同时可以 缓解传统服务器端压力过大的问题	难于管理,服务器的瘫痪 将导致整个网络的瘫痪
边缘缓存	需要快速及时获取数据的 地方,比如工厂的实时监控、 公司的数据管理等	降低了数据获取的延时;并且 用户可以从任何地方获取数据	需要定期将数据存放到云服务器, 容易造成数据的泄露;同时,缓存 容量有限,需要精心设计缓存放置策略

本文提出一种新的移动边缘网络模型,该模型由多个特殊设计的微边缘节点组成一个自组织边缘网络,协同为终端用户提供一跳无线服务。边缘节点协同预取和缓存来自云的数据内容,旨在提高终端用户的体验质量。主要难点是,如何为每个节点设计一个有效的分布式协作方案。通过引入经济学概念,提出了一个博弈模型来对每一个节点的行为进行建模。在博弈中,每个节点都以利润最大化为目标。实验表明:存在一个纳什均衡点,在该点上可以得到近似最优的缓存解。我们的主要研究结果如下。

- (1) 定义了一个合作的边缘缓存网络模型,该模型中有许多具有存储能力的边缘节点。用户接入和互联网接入形成了一个自组织网络,并且一个节点服务多个用户。移动用户的需求可以通过与其直接相连的边缘节点或通过边缘网络的其他边缘节点的缓存数据内容来满足。
- (2) 设计了一个非协作博弈模型来研究边缘节点之间的协作行为。在博弈中,每个边缘节点都是一个参与者,决定缓存(存储)初始化和替换策略。边缘节点的效用源于向移动用户发送缓存内容的数量和延时。通过最大化每个节点的收益,我们发现提出的博弈中存在纳什均衡,在均衡状态下,边缘节点的个体效用和边缘网络的全局效用同时近似最优。然后根据理论分析,设计了一种分布式均衡算法并仿真验证了其性能。

## 1 相关工作

近年来,边缘缓存问题一直是人们关注的焦点<sup>[28]</sup>。边缘缓存可以拉近内容与用户的距离,并提高服务质量(QoS),比如访问延时。同时,边缘缓存还可以节省骨干网络流量,提高边缘缓存的命中率。与边缘缓存相关的问题主要有以下 3 个方面:缓存内容、缓存位置以及缓存策略。

### 1.1 缓存内容

为了研究服务器等设备以及边缘设备该缓存哪些内容,很多研究集中在视频属性以及如何建模和预测视频的受欢迎程度上。通过对文件流行度的预测,节点可以判断出哪些内容更加受欢迎,从而将用户最需要的文件存放在硬件设备中。一项研究调查了在线视频的受欢迎程度和位置之间的关系<sup>[29]</sup>。这项研究发现:当用户来自一个有限的和单一地区时,视频展现出和兴趣有关的地理分布。事实上,这种视频测量方法可以用来描述单一区域的视频分布。但是如果大学等场所,学生在不同区域(例如教室和宿舍)的兴趣分布可能会不相同,因此得到的关于视频流行度的测量可能会不准确。基于早期流行度测量给出的历史信息的视频流行度

的预测也有相应的研究<sup>[30]</sup>。这种在线预测能够预测视频内容的受欢迎程度。由于这种方法没有视频需求的先验知识, 所以很难在一段时间内测量视频的受欢迎程度, 比如在高峰时间段的视频请求。还有一些研究人员关注离线缓存<sup>[31]</sup>, 他们让缓存工作在非高峰时间段, 它测量了用户在高峰时间对文件发出的请求数量, 并将其作为文件缓存的一个重要因素。离线缓存方案的优点是可以减少边缘设备的工作负载, 因为边缘设备仅仅缓存它已经更新过的内容。另外, 更加常用的一种方法是利用文件的统计特性, 例如齐夫分布<sup>[32]</sup>来表示文件流行度对缓存的影响。在本文中, 我们给出的解决方案主要是针对用户对文件的兴趣在短期内不会发生改变的情况, 例如宿舍、食堂、办公室等。

## 1.2 缓存位置

数据密集型应用对带宽的巨大需求给传统的内容交付形式带来了巨大的挑战, 所以为了提高用户的服务体验, 有人就将用户需要的内容从云中心下沉到边缘<sup>[33]</sup>。通过减少用户与内容之间的距离, 从而减少由于内容传输带来的延迟, 比如内容分发网络、内容提供商运营的 CDN<sup>[34]</sup>和对等 CDN<sup>[35]</sup>。在传统的云计算方案<sup>[36]</sup>中, 来自用户的每个请求都由云服务器提供。由于用户的请求在相同区域中基本上是相似的, 如果所有的请求都是由云服务器提供的, 那么在核心网络中就会产生大量的冗余流量, 从而对核心网络造成巨大的带宽压力。在网络边缘构建缓存是一种有效的解决方案<sup>[37]</sup>。通过对用户需求的常用内容进行缓存, 提出了一种设备到设备的缓存方案, 该方案很好地缓解了由于内容重复而造成的骨干网络流量紧张<sup>[38]</sup>。但是这种方案需要考虑中断概率, 这在多跳网络中很难实现, 即使是单跳 D2D 网络, 硬件的设计通常也比较困难。Hasti 等人<sup>[39]</sup>通过将内容缓存到更靠近用户的边缘接入点, 进一步降低了传统 CDN 中用户请求文件的时延, 提高了用户的服务体验。有学者使用博弈论方法对边缘缓存进行了研究<sup>[40]</sup>, 将边缘网络定义为一个一跳网络, 并分析了其延时特性。本文提出了一种新的边缘缓存模型, 在一组小单元基站中, 所有节点作为一个网络协同工作。不同于文献[38], 我们将边缘网络定义为一个多跳网络, 通过将边缘缓存问题建模为一个非协作博弈过程, 我们发现其延时特性和边缘系统的命中率要优于提出的其他方案。

## 1.3 缓存策略

由于用户的需求是不断变化的, 如何在缓存设备中进行文件的缓存存放策略和替换策略就显得十分重要。假设不同缓存上的请求是独立的, 我们可以通过一个分析模型<sup>[41]</sup>捕捉内容流行动态对缓存性能的影响。为了研究协同缓存, 提出了异构无线网络的编码缓存策略, 以平衡基站传输、接入点存储和用户连接延时之间的成本<sup>[7,42]</sup>。Rimal 等人<sup>[43]</sup>提出了一个集中式的边缘缓存方案, 并开发了概率生存分析模型来评估光纤切断和 MEC 服务器故障对边缘缓存性能的影响。但是这种集中式的方式不利于边缘系统的维护, 一旦核心边缘处理器出现故障, 整个系统将难以运行。为了优化系统功耗, 分布式缓存被提出来, 以减少平均视频传输延时<sup>[44]</sup>, 同时可以缓解核心处理器的计算压力。据我们所知, 在分布式边缘系统上的协同缓存很少被研究, 尤其是考虑到节点的收益来研究边缘系统的性能和设计一个有效的边缘内容交付的缓存策略。

总而言之, 边缘缓存的性能取决于精确的文件流行度的度量、文件在网络中的存储位置以及文件在网络中存储以及替换策略。因此, 在边缘缓存中有很多工作关注于视频测量<sup>[29-32]</sup>。另一些人关注于缓存位置的优化<sup>[33-40]</sup>, 关注最多的是缓存策略的设计<sup>[41-44]</sup>。

在我们的方案中, 我们考虑节点缓存文件带来的成本以及收益, 即节点如何在有限的缓存容量下做出使自己收益最大化的决策。另外, 我们假设用户的请求在一段时间内不会发生改变来简化我们的模型。

## 2 模型及问题公式化

在本节中, 考虑一个小型边缘缓存场景(例如餐厅、办公室或者家庭社区等)。在这种情况下, 节点之间的距离与节点与云计算单元之间的距离相比非常短。图 1 展示了一个简单的示例。

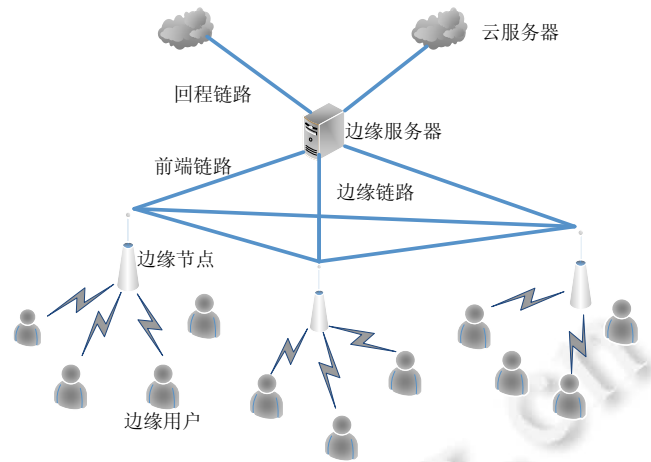


图1 边缘缓存网络模型

## 2.1 系统模型

在 5G 网络中, 由于其频率远高于 4G, 因此存在绕射能力更差、穿透能力更弱、覆盖范围更小的问题. 因此, 为了实现 4G 的全覆盖, 在 5G 的应用场景中则需要更多的基站. 因此, 在 5G 网络中, 大多采用部署小基站的方式, 这种小基站建设成本低, 可以大量部署, 甚至可以在每个家庭放置一个小基站, 从而实现更小范围的信号覆盖问题和 5G 的超密集组网. 同时, 它的组网方式灵活, 可以通过光纤、WLAN、网线等实现数据的回传.

考虑一个包含有  $N=\{1,2,\dots,i,\dots,N\}$  个节点的边缘网络, 由于小基站建设成本低, 因此其计算能力和缓存存储能力相对较低. 小基站通过回程链路连接到服务器. 边缘网络中的每个小基站可以缓存  $M=\{1,\dots,r,\dots,M\}$  个具有相同大小  $c$  的文件, 这些文件可以被边缘用户检索. 令  $w_i^r$  表示第  $i$  个节点上由用户产生的对第  $r$  个文件的请求量.

当一个用户发出的请求到达一个节点时, 节点首先从它自身的缓存中寻找该文件, 如果这个文件存在于该节点, 则该节点直接将文件传递给用户. 如果这个文件不存在于这个节点上, 则该节点向边缘网络中的其他节点寻找该内容, 如果其他节点有这个内容, 则节点会从距离它最近的节点向用户传递文件; 否则, 该节点将直接从云服务器向用户传递文件. 用户从本地节点、邻居节点以及云服务器获取同一个文件的最大的区别在于: 随着距离的增长, 用户的延迟也会逐渐增长, 因此用户的服务体验也会随之下降.

因此, 如何合理地在边缘节点上设计缓存策略, 不仅关系到用户满意度的最大化, 而且对减少延迟也十分重要. 传统的边缘缓存往往借助凸优化的方法, 通过最小化所有节点的总延时、骨干网络的流量或者最大化所有边缘节点的收益, 从而获得每个节点的缓存配置. 这样的方法的缺点在于: 它需要一个服务器为所有节点计算缓存配置, 对这个服务器的计算压力较大, 一旦这个服务器崩溃, 则整个边缘网络将不能再继续工作. 我们的方法优于传统方法的地方在于不需要一个服务器为每一个节点计算缓存配置, 每个节点单独计算, 这样的操作降低了单个服务器的计算压力, 同时, 单个服务器的损坏不会影响整个边缘网络的性能.

## 2.2 系统流程图

在本小节中, 对边缘缓存系统进行详细的描述, 具体的描述如图 2 所示. 5G 基站起着云服务器的作用, 它具有较大的容量, 服务器可以被所有节点访问. 家中的小基站则是边缘节点, 移动电话以及 Pad 则是边缘用户. 基于上述定义的系统模型, 我们的方案工作如下.

- 1) 第 1 步, 用户通过小基站将他们的设备连接到网络. 当用户向一个文件发出请求时, 他可以从小基站或者 5G 基站获取内容. 因为 5G 基站距离用户较远, 至少不会被部署在家里, 因此用户从基站获取内容的延时相比于用户从小基站获取内容的延时要高很多.

- 2) 第 2 步, 小基站会搜索自身的缓存. 如果小基站包含用户想要的内容, 则小基站就会将内容分发给用户; 如果在第 1 个连接到用户的小基站中没有找到内容, 则该小基站会从边缘网络的其他小基站搜索内容; 如果其他小基站有用户需要的内容, 则小基站会将该内容通过边缘网络发送给用户. 当小基站发现它需要的内容存在于边缘网络, 但是从边缘网络的其他小基站获取该内容的延迟超过从云服务器获取的延时, 则该小基站会直接从 5G 基站获取该内容.
- 3) 第 3 步, 当与发出请求的用户直接相连的小基站发现在边缘网络中没有用户需要的文件时, 则该小基站会直接从 5G 基站获取内容.

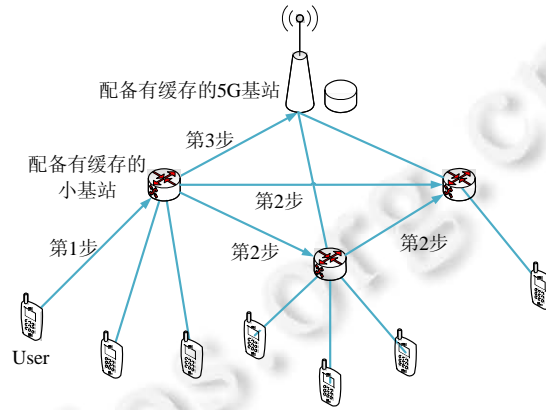


图 2 系统流程

### 2.3 问题建模

边缘网络拓扑用  $G=(N,g)$  表示, 其中,  $N=\{1,\dots,N\}$  表示边缘网络拥有的节点数量,  $g_{ij}$  表示是否节点  $i$  连接到节点  $j$ .  $l(i,j)$  表示节点  $i$  和节点  $j$  之间的最短跳数. 我们假设  $l(i,i)=0$ ,  $l(i,N+1)=D$  表示从云服务器获取内容的跳数,  $l(i,N+1)\gg l(i,j), \forall j \in N$ . 我们使用一系列的二进制变量  $a_i^r$  表示节点  $i$  是否缓存了文件  $r$ . 我们用  $\alpha$  表示节点从本地满足用户请求时的收益, 用  $\beta$  表示节点需要在一跳的距离获取内容的成本, 用  $\gamma$  表示一个文件需要从云服务器获取的成本. 这些参数在仿真过程中会用具体的数字代替. 一般的, 我们认为每个节点最多可以缓存  $p$  个文件, 并且  $\alpha < \beta < \gamma$ .

### 2.4 基本博弈模型

采用博弈论的方法来分析缓存放置问题. 在博弈论模型中有  $N$  个节点(玩家), 并且每个节点最多缓存  $p$  个文件. 在本文的其余部分, 我们将不区分节点和玩家. 节点的不同文件配置会给网络带来不同的延迟, 目标是优化节点的文件配置, 在其他节点的请求保持不变的情况下最小化延迟.

为了使用博弈论, 做出如下的解释. 在系统模型中有  $N$  个节点, 每个节点独立地做出缓存哪些文件的决策, 目的是最大化它的收益, 用  $A_i = \left\{ \sum_r a_i^r \leq p \right\} \subset \{0,1\}^M$  表示节点的缓存策略, 不等式表示缓存容量约束. 用  $a_i^r$  描述节点  $i$  是否缓存文件  $r$ . 用  $A_{-i}$  表示节点  $i$  以外其他节点的缓存策略. 用  $R_i(A_{-i})$  表示边缘网络中缓存了文件  $r$  的节点, 即  $R_i(A_{-i}) \stackrel{def}{=} \{n | a_i^r = 1, \forall n \in N\}$ . 由于本文关注的是离线缓存, 因此在流量高峰时期的白天, 节点上缓存的内容不随用户请求的不同而发生变化; 只有在流量不繁忙的夜晚, 节点才会根据不同文件的请求量对自身的缓存进行更新. 到目前为止, 可以定义一个节点的效用模型(即一个节点的总货币收益)如下.

- 如果  $a_i^r = 0$ :

$$C_i^r(\{a_i^r\}, A_{-i}) = \begin{cases} w_i^r(\gamma - \alpha), & a_i^r \notin R_i(A_{-i}) \\ w_i^r(\beta l(i, j) - \alpha), & a_i^r \in R_i(A_{-i}) \end{cases} \quad (1)$$

- 如果  $a_i^r = 1$ :

$$C_i^r(\{a_i^r\}, A_{-i}) = \begin{cases} w_i^r \alpha + \sum_{k \in N \setminus \{i\}} w_k^r \beta l(i, k), & a_i^r \notin R_i(A_{-i}) \\ w_i^r \alpha + \sum_q w_q^r \beta l(i, q) + \sum_s w_s^r \beta l(i, s), & a_i^r \in R_i(A_{-i}) \end{cases} \quad (2)$$

使用  $C_i^r(A_i, A_{-i})$  来表示节点  $i$  的效用,  $q = \{l(i, j) - l(R_i(A_{-i}), q) = 0 | q \in N \setminus (\{i\} \cup R_i(A_{-i}))\}$ ,  $s = \{l(i, s) - l(R_i(A_{-i}), s) = 0 | s \in N \setminus (\{i\} \cup R_i(A_{-i}))\}$ . 在公式(1)中,  $\gamma - \alpha$ 指的是在边缘网络中没有任何一个边缘节点缓存文件  $r$  的情况下, 此时如果用户向与他直接相连的节点请求内容时, 首先用户会向该节点“支付” $\alpha$ 的成本, 即节点会收到来自用户  $\alpha$ 的收益, 但由于整个边缘网络都没有该用户需要的内容, 因此该节点为了满足用户的请求, 就需要向云服务器获取文件  $r$ , 那么在这个过程中, 该节点就需要向云服务器支付  $\gamma$ 的费用, 即节点的成本. 综上所述, 节点在这种情况下的一次收益即为  $\gamma - \alpha$ .

用下面的方法将节点的收益分成以下 4 个部分, 分别用  $p_1, p_2, p_3, p_4$  表示这 4 部分的子收益.

- 1) 当文件  $r$  缓存在节点  $i$  和其他节点. 如果有用户需要文件  $r$ , 则节点  $i$  的子收益  $p_1$  可以表示为

$$p_1 = \sum_{a_i^r \in R_i(A_{-i})} \left( w_i^r \alpha + \sum_q w_q^r \beta l(i, q) + \sum_s w_s^r \beta l(i, s) \right) \quad (3)$$

- 2) 当文件  $r$  缓存在其他节点而没有缓存在节点  $i$ , 如果有用户需要文件  $r$ , 则节点  $i$  的子收益  $p_2$  可以表示为

$$p_2 = \sum_{a_i^r \in R_i(A_{-i})} w_i^r (\beta l(i, j) - \alpha) \quad (4)$$

- 3) 当文件  $r$  没有缓存在其他节点, 但是缓存在节点  $i$ , 如果有用户需要文件  $r$ , 则节点  $i$  的子收益  $p_3$  可以表示为

$$p_3 = \sum_{a_i^r \notin R_i(A_{-i})} a_i^r (w_i^r \alpha) + \sum_{k \in N \setminus \{i\}} w_k^r \beta l(i, k) \quad (5)$$

- 4) 当文件  $r$  没有缓存在边缘网络, 如果有用户需要文件  $r$ , 则节点  $i$  的子收益可以表示为

$$p_4 = \sum_{a_i^r \notin R_i(A_{-i})} w_i^r (\gamma - \alpha) \quad (6)$$

因此, 节点  $i$  的总收益可以很方便地表示为如下的形式:

$$u_i(A_i, A_{-i}) = \sum_r C_i^r(a_i^r, A_{-i}) = p_1 + p_2 + p_3 + p_4 \quad (7)$$

等式(7)表达了节点  $i$  的总收益. 总收益表明了 4 种不同情况下的收益之和, 这 4 种情况下, 对所有文件在某一个节点上的收益不重复不遗漏, 分别是: 文件  $r$  缓存在节点  $i$  上, 同时缓存在边缘网络中的其他节点, 即收益  $p_1$ ; 文件  $r$  没有缓存在节点  $i$  上, 但是缓存在其他节点上, 即收益  $p_2$ ; 文件  $r$  缓存在节点  $i$  上, 但是没有缓存在边缘网络中的其他节点上, 即收益  $p_3$ ; 文件  $r$  没有缓存在节点  $i$  上, 同时也没有缓存在边缘网络中的其他节点上, 即收益  $p_4$ . 从而就可以定义一个策略式博弈  $\Gamma(N, (A_i)_{i \in N}, (u_i)_{i \in N})$ . 其中, 设备集合  $N$  表示博弈中的玩家集合.  $A_i$  表示玩家  $i$  受到缓存大小约束的策略空间.  $u_i$  表示玩家  $i$  的基于路径远近的效用. 节点  $i$  的目标是通过调整  $A_i$  最大化它的收益. 表 2 总结了本文使用的关键符号.

表 2 关键符号

名称	描述
$N$	边缘节点
$M$	总的文件数
$w_i^r$	节点 $i$ 产生的对文件 $r$ 的请求数
$G=(N, g)$	边缘网络拓扑
$g_{ij}$	节点 $i$ 是否与节点 $j$ 相连
$l(i, j)$	节点 $i$ 和节点 $j$ 之间的最短跳数
$a_i^r$	节点 $i$ 是否缓存文件 $r$

表 2 关键符号(续)

名称	描述
$\alpha$	节点从本地满足用户请求的收益
$\beta$	节点从一跳的范围内获取文件的成本
$\gamma$	从云服务器获取文件的成本
$A_i$	节点 $i$ 的缓存策略
$R_r(A_i)$	缓存了文件 $r$ 的节点集合
$u_i(\cdot)$	节点 $i$ 的效用

### 3 协作缓存中的纳什均衡

在本节中, 我们为所提出的边缘缓存方案设计了一个分布式的算法. 在本文中, 节点之间的“协作”的意义在于: 在网络流量压力较小的时间段, 比如夜间, 网络边缘的节点可以通过广播的方式将自身的缓存内容告知网络中的其他节点. 这样, 网络中的所有的节点在进行缓存决策时就会充分考虑其他节点的缓存内容, 从而能够更加合理地缓存自身的内容. 为了设计这个算法, 首先证明了纳什均衡的存在性.

**定义 1(纳什均衡).** 令  $a^* \triangleq [a_1^*, a_2^*, \dots, a_N^*]$  是博弈  $\Gamma$  的解. 如果对于博弈  $\Gamma$ , 下面的不等式成立:

$$u_i(a_i^*, a_{-i}^*) \geq u_i(a_i, a_{-i}^*), \forall m, a_i \in A_i \tag{8}$$

在公式(8)中,  $m$  指的是参与博弈的任意一个节点, 它的缓存内容都需要满足这个不等式.

**定理 1(Nash 1951).** 具有有限数量的玩家和有限数量的行动集合的博弈至少存在一个纳什均衡.

为了寻找到我们所提出的边缘缓存方案的均衡解, 我们作如下考虑: 首先定义一个无向图  $G$ , 无向图  $G$  中每一个节点代表一个可能的纯策略缓存的集合. 当网络中除了玩家(节点) $i$  的其他玩家(节点)确定了他们的缓存策略之后, 玩家  $i$  需要确定一个对于其他玩家策略给定之后的最佳策略, 这个策略可以使玩家  $i$  在其他玩家策略不变的情况下, 使玩家  $i$  的收益达到最大.

**定义 2(最佳响应).** 玩家  $i$  对应于策略集合  $a_{-i}$  的最佳响应是策略集  $a_i^*$ , 对于玩家  $i$  所有可选的策略集合  $a_i \in A_i$ , 满足  $u_i(a_i^*, a_{-i}) \geq u_i(a_i, a_{-i})$ .

无向图中, 从一个节点替换成另一个节点需要满足如下不等式:

$$\{a_1, a_2, \dots, a_i, \dots, a_N\} < \{a_1, a_2, \dots, a_i', \dots, a_N\} \tag{9}$$

当所有的玩家都满足上述不等式的时候, 我们就说  $\{a_1, a_2, \dots, a_i, \dots, a_N\}$  是一个纯策略纳什均衡. 不等式中没有等号的原因在于: 我们认为, 当一个节点通过替换策略没有使自身的效用增加的时候, 我们可以不改变该节点的缓存内容. 通过让每个节点都能达到最佳响应, 从而寻找到均衡.

我们对提出的算法做如下解释: 在流量不繁忙的夜间, 边缘节点开始根据网络中用户积累的内容请求量的大小以及其他节点的缓存内容, 开始更新自身的缓存, 目的是使得在不同的缓存配置下, 节点能够寻找到使得自身收益最大的缓存配置. 节点首先计算出缓存每一个文件为它带来的收益, 然后选择在满足缓存容量限制下的文件作为其缓存内容, 然后, 该节点将它缓存的内容广播, 目的是使得边缘网络中其他节点能够通过它所更新的缓存内容从而重新选择使它们收益最大的缓存配置. 以上的操作会逐个计算, 直到所有节点在其他节点不改变内容的情况下, 自身也不会改变内容为止, 缓存的更新操作结束. 我们算法的实现细节见算法 1.

**算法 1.** 基于博弈论的协作缓存放置算法.

输入:  $k, \alpha, \beta, \gamma, w_r^j, l(i, j)$ .

输出:  $a_i^j$ .

初始化:  $a_i^j = 0, u_i(A_i, A_{-i}) = 0, i = 1$

1: **for**  $i < N$  **do**

2:  $m_r = u_i(a_i^r = 1, A_{-i}) - u_i(a_i^r = 0, A_{-i}), \forall r \in [1, j]$ , 按降序排序  $m_r$  并且更新  $a_i^r$



```

3:   if  $a_i^r = a_i^{r'}$  then
4:      $k++$ 
5:   else
6:      $k=1$ 
7:   end if
8: end for
9: if  $k \neq N$  then
10: 重复步骤 3-步骤 8
11: end if
12: else break
13: return  $a_i^r$ 

```

#### 4 仿真分析

为了评价该算法的有效性,我们进行了实验研究,利用我们的算法给出了文件在小型网络中的配置情况,并计算分析了其在命中率、延迟和流量减少方面做出的贡献.为了方便起见,我们为初始参数设置一些特定的值.我们考虑:如果用户可以从直接相连的节点获得文件,则用户向该节点支付 10;如果与用户直接相连的节点没有用户需要的文件,则该节点会检索邻居节点的文件;如果邻居节点有用户需要的节点,则该节点会将内容通过边缘网络传递给用户,用户会向每跳网络支付 20;最后,如果边缘网络没有节点拥有该用户需要的内容,则用户会向云服务器寻求内容,则用户会向云服务器支付 80.此外,我们设置边缘节点的容量为 5,总文件数为 50.虽然上述假设在真实的网络环境中会发生变化,但是我们的设置不会影响算法的性能,其主要的原因在于:基于非协作博弈的方案能够更好地选择“缓存哪些内容”,即使节点的数据增加,节点选择缓存内容的算法是不会改变的.

##### 4.1 边缘节点数量对边缘命中率和骨干网络流量的影响

边缘节点的数量决定了边缘网络缓存池的大小,边缘命中率指的是用户可以在边缘得到的请求与边缘网络所有用户的所有请求之比.显然,边缘网络的缓存池越大,用户可以在边缘获取到文件的可能性也会增大.我们将我们的方案与贪婪算法下的性能进行比较,贪婪算法下的节点不会考虑其他节点的缓存可能会为自己带来收益,仅仅考虑请求量最多的文件作为自身的缓存.图 3 显示了通过增加边缘节点数量,不同的缓存方法对缓存命中率的影响.

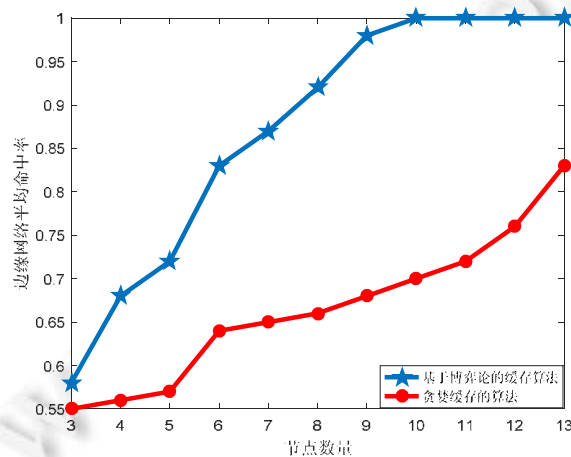


图 3 节点数量的变化对边缘命中率的影响

从图 3 中我们可以看到, 随着边缘节点数量的增加, 我们的方案和贪婪算法下的平均内容命中率都在增加. 平均命中率的增加是因为随着缓存空间的增加, 可以通过边缘网络存储和共享更多的内容. 由于基于非协作缓存的缓存方案考虑了整个边缘网络的需求, 因此这种方法可以获得更高的边缘命中率. 但由于人们在小区域具有相似的兴趣, 基于贪婪算法的文件缓存会导致大量的内容冗余. 通过模拟, 我们发现: 我们的方案在边缘节点的总容量超过云空间时, 边缘命中率会达到 100%.

同样地, 如果节点可以在边缘网络缓存更多的文件, 则可以节省更多的骨干网络的流量. 图 4 显示了边缘网络的大小对骨干网络流量的影响. 边缘节点的数量越多, 骨干网络流量节省的就越多. 当边缘节点的总的缓存池超过系统总文件数的大小时, 平均节省的骨干网络的流量性能收敛到 1. 这表明, 我们的算法在小区域内具有很好的性能.

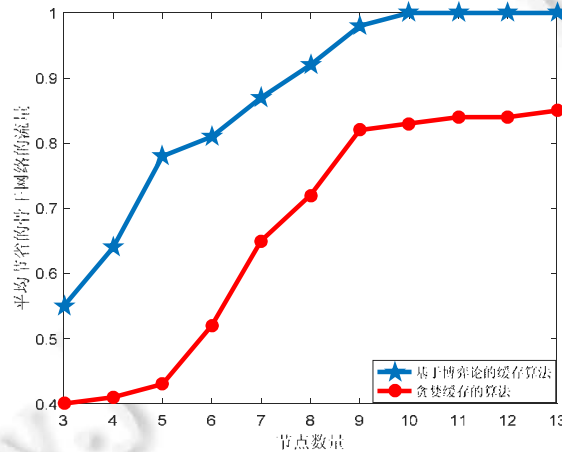


图 4 节点数量变化对骨干网络流量的影响

#### 4.2 系统关键参数对缓存命中率的影响

在图 5 中, 我们研究了改变系统关键参数对边缘命中率的影响. 如上所述: 我们假设从直接相连的节点获得内容会花费  $\alpha$ , 从边缘节点获得文件花费  $\beta$ , 从云服务器获得文件花费  $\gamma$ . 我们给定  $\alpha=10$ ,  $\gamma=80$ , 改变一跳的成本  $\beta$  从 20 到 50. 从图中我们可以观察到: 在非协作博弈中, 节点具有最高的一跳命中率. 我们还观察到: 随着单跳成本的增加, 节点越来越倾向于从云服务器获取内容, 而不是从具有多跳的其他节点. 因为在非合作博弈方案中, 基站的目标是使其收益最大化. 如果节点从相邻节点检索文件的成本高于从云服务器检索文件, 则该节点会直接从云服务器向用户提供服务. 另一方面, 随着边缘命中率的下降, 为了降低成本, 核心网络的流量明显增加了. 由此可见, 系统的关键参数对整个边缘网络的影响是显著的. 因此, 实现边缘节点之间的协作是至关重要的, 这在之前的很多工作中都是没有完成的.

在图 6 中, 我们研究了在考虑和不考虑节点成本的情况下, 缓存策略对缓存命中率的影响. 首先, 我们观察到: 在不考虑节点成本的情况下, 随着一跳成本的增加, 协作缓存方案的命中率几乎保持不变. 其次, 我们观察到: 在考虑边缘成本的协作缓存方案中, 随着单跳成本的增加, 边缘缓存的命中率在下降. 这是因为随着单跳成本的增加, 如果节点想要保持较高的命中率, 节点就必须花费更多的成本从边缘网络获取内容, 从而使得成本高于从云服务器获取内容.

在图 7 中, 我们研究了考虑和不考虑边缘命中率对边缘节点平均收益的影响. 首先, 我们观察到: 在不考虑边缘命中率的情况下, 随着一跳成本的增加, 边缘节点的平均收益也在下降. 其次, 我们发现: 在考虑边缘命中率的情况下, 边缘节点的平均收益下降得更快. 这是因为随着单跳代价的增加, 节点需要向相邻的节点支付更多的费用; 另一方面, 考虑到边缘命中率, 为了达到更高的边缘命中率, 节点可能需要花费比从核心网络检索内容更大的代价.

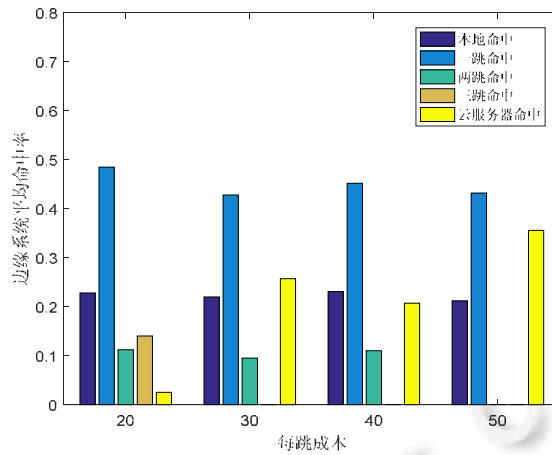


图5 系统关键参数对缓存命中率的影响

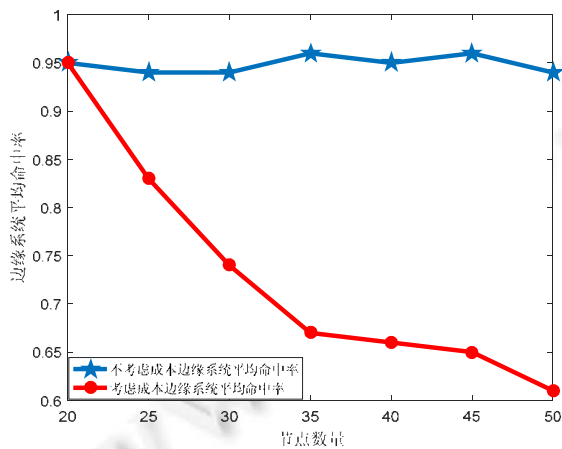


图6 命中率优先

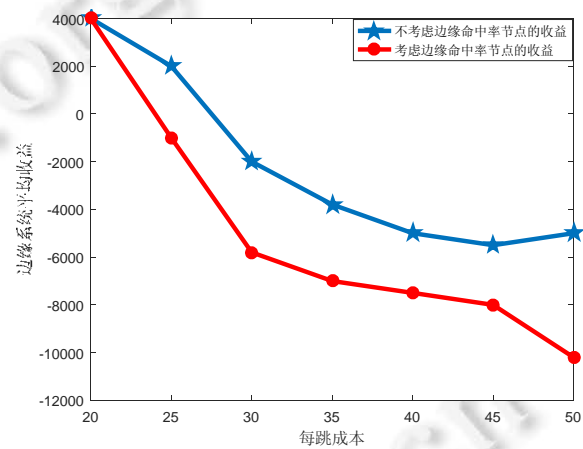


图7 收益优先

通过以上对系统关键参数的分析我们发现，随着成本的增加，如果系统想要获得更高的命中率，降低核心网络的带宽，就要以牺牲效益为代价；如果边缘网络希望自己的收益最大化，那么就会增加核心网络的流量。因此，选择合适的系统参数对边缘缓存系统获得良好的网络性能具有重要意义。为了在命中率和代价之间取得平衡，我们可以通过设置合理的一跳成本，在不降低节点命中率的情况获得更高的收益。

#### 4.3 与全局最优化方法的比较

在图 8 中，由于我们的目标是在非合作博弈的框架中最大化节点的利益，所以我们将全局最优化的缓存方案和基于非协作博弈的方案进行了比较，我们选择系统参数为  $\alpha=10$ ,  $\beta=20$ ,  $\gamma=80$ 。我们观察到：在全局最优的方案中，系统可以获得最大的边缘缓存效益。我们的方案要弱于全局最优解，这是因为我们的方案是分布式的，它没有直接优化全局收益。但是分布式方案的好处在于节点可以独立地进行缓存决策，这对于系统设计非常方便。

在图 9 中，我们比较了全局最优解和我们的方案对边缘系统命中率的影响。在命中率方面，我们的方案明显优于全局最优的方案。这是因为尽管我们的方案是分布式的，但是每当一个节点做出缓存决策时，它都会考虑其他节点的缓存内容，从而最优化自身的内容放置。

通过与全局最优解的比较，虽然我们的系统收益不如全局最优方案，但是在边缘命中率方面，我们的方案的性能明显优于全局最优解。

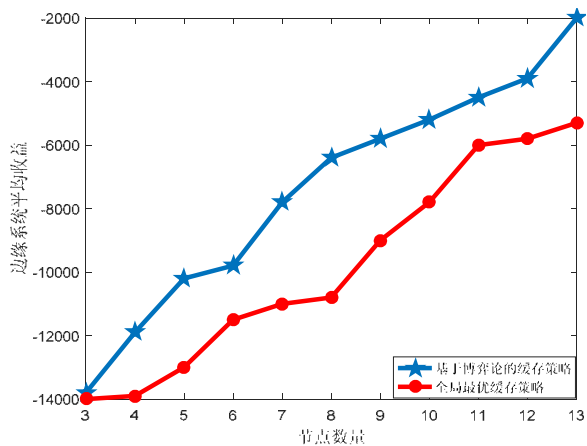


图8 与全局最优策略的成本比较

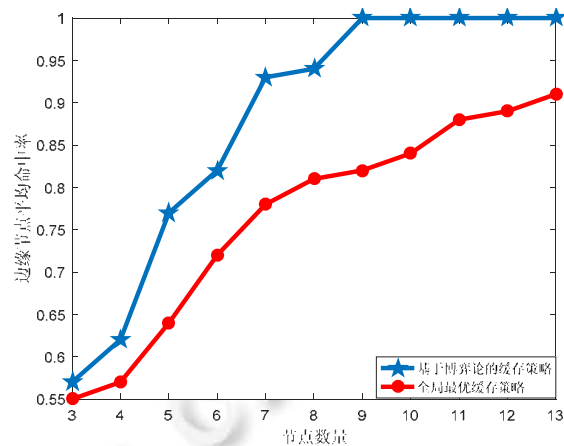


图9 与全局最优策略的边缘命中率比较

## 5 结论

在本文中,我们将设备之间的协作建模为一个非协作博弈.我们设计了一个缓存放置算法,并且证明了纳什均衡的存在性,通过仿真验证了该方法的有效性.结果表明:与贪婪缓存算法相比,我们所提出的算法在平均边缘命中率方面具有更好的性能.此外,随着边缘节点的增加,整个边缘网络的缓存命中率也在增加.在用户需求快速增长的今天,由于采用了大数据、机器学习等新技术,人们对用户需求的预测也越来越准确.因此可以根据我们的算法更好地制定缓存方案,在网络的非高峰时期进行缓存,并在网络的高峰时期通过边缘网络满足用户请求,从而降低核心网络的流量.在今后的研究中,我们将研究如何在我们考虑的系统中保证物理层安全<sup>[45-47]</sup>并且使用SE(可搜索加密, searchable encryption)<sup>[48,49]</sup>.此外,本文所考虑的缓存策略如何应用在IIoT<sup>[50,51]</sup>场景中,还有待进一步研究.

## References:

- [1] Christopoulos D, Chatzinotas S, Ottersten B. Cellular-broadcast service convergence through caching for CoMP cloud RANs. In: Proc. of the IEEE Symp. on Communication and Vehicular Technology in the Benelux (SCVT). IEEE, 2015. 1-6.
- [2] Ejder B, Bennis M, Debbah M. Living on the edge: The role of proactive caching in 5G wireless networks. IEEE Communication Magazine, 2014, 52(8): 82-89.
- [3] Wang Y, Xu J, Jiang LS. Challenges of system-level simulations and performance evaluation for 5G wireless networks. IEEE Access, 2014, 2: 1553-1561.
- [4] Crovella ME, Azer B. Self-similarity in World Wide Web traffic: Evidence and possible cases. IEEE/ACM Trans. on Networking, 1997, 5(6): 835-846.
- [5] Cha MY, Haewoon K, Pablo R, et al. I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system. In: Proc. of the 7th ACM SIGCOMM Conf. on Internet Measurement. IEEE, 2007. 1-14.
- [6] Asadi B, Ong L. Centralized caching with shared caches in heterogeneous cellular networks. In: Proc. of the IEEE 20th Int'l Workshop Signal Processing Advanced Wireless Communication (SPAWC). IEEE, 2019. 1-5.
- [7] Maddah-Ali MA, Niesen U. Fundamental limits of caching. IEEE Trans. on Information Theory, 2014, 60(5): 2856-2867.
- [8] Wang D, Lan Y, Zhao T, et al. On the design of computation offloading in cache-aided D2D multicast networks. IEEE Access, 2018, 6: 63426-63441.
- [9] Zhou K, Sun S, Wang H, et al. Improving cache performance for large-scale photo stores via heuristic prefetching scheme. IEEE Trans. on Parallel Distributed System, 2019, 30(9): 2033-2045.
- [10] Wei Y, Banawan K, Ulukus S. Cache-Aided private information retrieval with partially known uncoded prefetching: Fundamental limits. IEEE Journal on Selected Areas in Communications, 2016, 35(2): 380-391.
- [11] Wang XF, Chen M, Tarik T, et al. Cache in the air: Exploiting content caching and delivery techniques for 5G system. IEEE Communication Magazine, 2014, 36(6): 1126-1139.

- [12] Haig S. The business case for caching in 4G LTE networks. LSI-wireless Technical Report, 2012, 10(3): Article No.43.
- [13] Valentino P, Gyrgy D. Distributed caching algorithms for interconnected operator CDNs. *IEEE Journal on Selected Areas in Communications*, 2017, 35(2): 380–391.
- [14] Ding P, Wei X, Dong Z, *et al.* A cache replacement strategy for hybrid P2P VOD streaming media. In: Proc. of the IEEE Int'l Conf. in Communication. IEEE, 2017. 1–6.
- [15] Xie W, Wang X. The improvement of cache replacement algorithm in P2P streaming media system. In: Proc. of the 4th Int'l Conf. on Systems and Informatics (ICSAI). IEEE, 2017. 949–953.
- [16] Panahi FH, Ohtsuki T. Energy efficiency analysis in cache-enabled D2D-aided heterogeneous cellular networks. *IEEE Access*, 2020, 8: 19540–19554.
- [17] Chen Y, Liu Y, Zhao J, *et al.* Mobile edge cache strategy based on neural collaborative filtering. *IEEE Access*, 2020, 8: 18475–18482.
- [18] Li XH, Wang XF, Victor L. Weighted network traffic offloading in cache-enabled heterogeneous networks. In: Proc. of the IEEE Int'l Conf. on Communications (ICC). IEEE, 2016. 1–6.
- [19] Marios K, Mehdi B, Merouane D. On the delay of geographical caching methods in two-tiered heterogeneous networks. In: Proc. of the IEEE 17th Int'l Workshop on Signal Processing Advanced in Wireless Communications (SPAWC). IEEE, 2016. 1–5.
- [20] Wang R, Peng X, Zhang J, Letaief KB. Mobility-Aware caching for content-centric wireless networks: Modeling and methodology. *IEEE Communication Magazine*, 2016, 54(8): 77–83.
- [21] Ying LZ, Cao GH. Supporting cooperative caching in ad hoc networks. *IEEE Trans. on Mobile Computing*, 2006, 5(1): 77–89.
- [22] Konstantinos P, Leandros T. Code, cache and deliver on the move: A novel caching paradigm in hyper-dense small-cell networks. *IEEE Trans. on Mobile Computing*, 2017, 16(3): 675–687.
- [23] Zhou S, Gong G, Zhou ZY, *et al.* GreenDelivery: Proactive content caching and push with energy harvesting based small cells. *IEEE Communication Magazine*, 2015, 53(4): 142–149.
- [24] Peng Y, Zhang N, Bi YG, *et al.* Catalyzing cloud-fog interoperation in 5G wireless networks: An SDN approach. *IEEE Network*, 2017, 31(5): 14–20.
- [25] Wang K, Yu FR, Li HY. Information-Centric virtualized cellular network with device-to-device (D2D) communications. *IEEE Trans. on Vehicle Technology*, 2016, 31(5): 14–20.
- [26] Hao H, Xu C, Wang M, *et al.* Stochastic cooperative multicast scheduling for cache-enabled and green 5G networks. In: Proc. of the IEEE Int'l Conf. on Communications (ICC). IEEE, 2019. 1–6.
- [27] Desai SK, Dua A, Kumar N, *et al.* Cache poisoning prevention scheme in 5G-enabled vehicular networks: A tangle-based theoretical perspective. In: Proc. of the IEEE 17th Annual Consumer Communications and Networking Conf. (CCNC). IEEE, 2020. 1–6.
- [28] Zhang KY, Gui XL, Ren DW, *et al.* Survey on computation offloading and content caching in mobile edge networks. *Ruan Jian Xue Bao/Journal of Software*, 2019, 30(8): 2491–2516 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/5861.htm> [doi: 10.13328/j.cnki.jos.005861]
- [29] Ander B, Salvatore S, Mirjam W. YouTube around the world: Geographic popularity of videos. In: Proc. of the 21st Int'l Conf. on World Wide Web. 2012. 241–250.
- [30] Xu J, Schaar D, Liu JC, *et al.* Forecasting popularity of videos using social media. *IEEE Journal on Selected Areas in Communications*, 2014, 9(2): 330–343.
- [31] Park J, Lee J, Namgung E, *et al.* A cache management scheme to support offline environment in cloud system. In: Proc. of the 4th IEEE Int'l Conf. on Cloud Computing Technology and Science (CloudCom). IEEE, 2012. 397–392.
- [32] Xiong ZH, Feng SH, Niyato D, *et al.* Game theoretic analysis for joint sponsored and edge caching content service market. In: Proc. of the IEEE Global Communication Conf. IEEE, 2018. 1–7.
- [33] Poularakis K, Iosifidis G, Argyriou A, *et al.* Video delivery over heterogeneous cellular networks: Optimizing cost and performance. In: Proc. of the IEEE Global Communication Conf. IEEE, 2016. 1078–1086.
- [34] Wang Z, Chen W, Sun MH, *et al.* CPCDN: Content delivery powered by content and user intelligence. *IEEE Trans. on Multimedia*, 2014, 17(2): 92–103.
- [35] Zhang G, Liu W, Hei XJ, *et al.* Unreeling Xunlei Kankan: Understanding hybrid CDN-P2P video-on-demand streaming. *IEEE Trans. on Multimedia*, 2014, 17(2): 229–242.
- [36] Jamal F, Abdullah MT, Hanapi ZM, *et al.* Reliable access control for mobile cloud computing (MCC) with cache-aware scheduling. *IEEE Access*, 2019, 7: 165155–165165.

- [37] James R, Nada S. Exploring the memory-bandwidth tradeoff in an information-centric network. In: Proc. of the 25th Int'l Teletraffic Congress (ITC). 2013. 1–9.
- [38] Cai J, Xu X, Liu Y, *et al.* Network coding-based socially aware caching strategy in D2D. *IEEE Access*, 2020, 8: 12784–12795.
- [39] Hasti A, Sujit D. Video aware scheduling and caching in the radio access network. *IEEE Trans. on Network*, 2014, 22(5): 1444–1462.
- [40] Wang C, Gao L, Gao QL, *et al.* Research on data cooperative cache strategy based on game theory in edge computing. *Application Research of Computers*, 2020, 37(12): 1–6 (in Chinese with English abstract).
- [41] Michele G, Emilio L, Stefano T. Efficient analysis of caching strategies under dynamic content popularity. In: Proc. of the IEEE Conf. on Computer Communications (INFOCOM). IEEE, 2015. 2263–2271.
- [42] Jad H, Nikhil K, Suhas D. Content caching and delivery over heterogeneous wireless networks. In: Proc. of the IEEE Conf. on Computer Communications (INFOCOM). IEEE, 2015. 756–764.
- [43] Rimal BP, Van DP, Maier M. Mobile-edge computing versus centralized cloud computing over a converged FiWi access network. *IEEE Trans. on Network and Service Management*, 2017, 14(3): 498–513.
- [44] Konstantinos P, George I, Antonios A, *et al.* Caching and cooperative policies for layered video content delivery. In: Proc. of the IEEE Conf. on Computer Communications (INFOCOM). IEEE, 2016. 1–9.
- [45] Li X, Zhao M, Gao X, *et al.* Physical layer security of cooperative NOMA for IoT network under I/Q imbalance. *IEEE Access*, 2020, 8: 51189–51199.
- [46] Shu F, Shen T, Xu L, *et al.* Directional modulation: A physical-layer security solution to 5G and further wireless networks. *IEEE Network*, 2020, 34(2): 210–206.
- [47] Balakrishnan S, Gupta S, Bhuyan A, *et al.* Physical layer identification based on spatial-temporal beam features for millimeter wave wireless networks. *IEEE Trans. on Information Forensics and Security*, 2010, 15(3): 1831–1845.
- [48] Wang H, Ning J, Huang X, *et al.* Security fine-grained encrypted keyword search for e-healthcare cloud. *IEEE Trans. on Dependable and Secure Computing*, 2019, 16(3): 1–14.
- [49] Chen B, Wu L, Li L, *et al.* A parallel and forward private searchable public-key encryption for cloud-based data sharing. *IEEE Access*, 2020, 8: 28009–28020.
- [50] Yang B, Cao X, Li X, *et al.* Mobile edge computing based hierarchical machine learning tasks distributed for IIoT. *IEEE Journal of Internet Things*, 2020, 7(3): 2169–2180.
- [51] Hou X, Ren Z, Yang K, *et al.* IIoT-MEC: A novel mobile edge computing framework for 5G-enable IIoT. *IEEE Journal of Internet Things*, 2020, 7(3): 1–7.

#### 附中文参考文献:

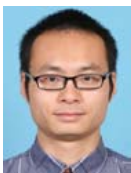
- [28] 张开元, 桂小林, 任德旺, 等. 移动边缘网络中计算迁移与内容缓存研究综述. *软件学报*, 2019, 30(8): 2491–2516. <http://www.jos.org.cn/1000-9825/5861.htm> [doi: 10.13328/j.cnki.jos.005861]
- [40] 王朝, 高岭, 高全力, 等. 边缘计算中基于博弈论的数据协作缓存策略研究. *计算机应用研究*, 2020, 37(12): 1–6.



顾汇贤(1996—), 男, 博士生, 主要研究领域为边缘计算, 无线通信, 博弈论.



魏贵义(1973—), 男, 博士, 教授, 主要研究领域为计算机网络, 信息安全, 分布式计算.



王海江(1987—), 男, 博士, 讲师, 主要研究领域为属性基加密, 可搜索加密.