

一种移动内容分发网络的分层协同缓存机制

葛志诚¹⁾ 徐 恪¹⁾ 陈 靛²⁾ 李 彤¹⁾ 姚 龙¹⁾ 沈 蒙³⁾

¹⁾(清华大学计算机科学与技术系 北京 100084)

²⁾(中国石油勘探开发研究院 北京 100083)

³⁾(北京理工大学计算机学院 北京 100081)

摘 要 近年来,随着网络流量的急剧增长,边缘计算(Edge Computing)成为降低网络负载、提高网络容量和服务质量的重要手段.移动内容分发网络(Mobile Content Delivery Network, MCDN)应运而生,已被通信运营商逐步采用.移动 CDN 将缓存节点建立在移动核心网络和无线接入网中;用户之间通过移动网络共享缓存空间,进行协同缓存.与传统 CDN 相比,移动 CDN 新增接入网侧基站协作缓存层和用户侧协作缓存层.在该系统中,每层缓存层采用何种缓存放置和替换策略,多层缓存之间如何协调配合,这一问题当前尚无确切解决方案.该文从此问题着手,对移动 CDN 分层协作缓存系统建立以最小化用户访问资源的整体代价为目标的最优化模型,并以此模型提出一种基于效用的启发式分层协作缓存策略.该策略根据待缓存资源全局效用值进行缓存决策,最小化用户资源访问代价.仿真结果和基于真实网络数据的验证实验表明,该策略能够有效降低用户资源访问代价,提高缓存命中率.

关键词 移动内容分发网络;边缘计算;分层协作缓存;缓存替换策略;D2D

中图法分类号 TP393 **DOI 号** 10.11897/SP.J.1016.2018.02769

A Hierarchical Cooperative Caching Strategy for Mobile Content Delivery Network

GE Zhi-Cheng¹⁾ XU Ke¹⁾ CHEN Liang²⁾ LI Tong¹⁾ YAO Long¹⁾ SHEN Meng³⁾

¹⁾(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

²⁾(PetroChina Research Institute of Petroleum Exploration & Development, Beijing 100083)

³⁾(School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081)

Abstract With the rapid growth of network traffic in recent years, edge computing has become an important method to reduce network load, increase network capacity and improve service quality. In this context, mobile content delivery network (mobile CDN) has been gradually adopted by ISPs. Within a mobile CDN, cache nodes are deployed in the mobile network, i. e., the radio access network (RAN). In the LTE network, the cache nodes in RAN sink to the base station. The main feature of RAN cache is that the single node only covers a small number of users; the storage capacity of a single node is limited, while there are many nodes in RAN. Therefore, it is necessary to perform cooperative caching. Besides, users share their cache space through wireless network by cooperative caching. Compared with traditional CDN system, the mobile CDN newly adds a cooperative caching layer in RAN and a cooperative layer on user side. So far, there is no exact

收稿日期:2017-07-16;在线出版日期:2018-10-12. 本课题得到国家重点研发计划(2018YFB0803405)、国家自然科学基金(61472212)、国家科技重大专项(2015ZX03003004)资助. 葛志诚,男,1993年生,硕士,中国计算机学会(CCF)学生会员,主要研究方向为物联网和内容分发网络. E-mail: gzc12345@qq.com. 徐 恪,男,1974年生,博士,教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为互联网体系架构、高性能路由器、P2P网络、物联网和网络经济学. 陈 靛,女,1972年生,网络与信息安全资深专家,主要研究方向为互联网体系架构、网络攻防技术、物联网与工控安全、密码学. 李 彤,男,1989年生,博士,中国计算机学会(CCF)学生会员,主要研究方向为移动云计算和传输协议. 姚 龙(通信作者),男,1976年生,博士研究生,中国计算机学会(CCF)学生会员,主要研究方向为物联网和信息中心网络. E-mail: yaolong10@mails. tsinghua. edu. cn. 沈 蒙,男,1988年生,博士,讲师,中国计算机学会(CCF)会员,主要研究方向为物联网和隐私保护.

solution on which caching allocation and replacement strategy should be used and how to coordinate between multi-layer caches in mobile CDN system. In this paper, we establish an optimization model for mobile CDN hierarchical cooperative cache system, aiming to minimize the overall link cost of user access to resources. Based on this model, we proposed a heuristic hierarchical utility-based cooperative caching (UCC) strategy based on real user demands. UCC makes cache decisions according to the global utility value of the cache resource to reduce the cost of resource access as much as possible. Simulation results and trace-based verification experiments show that the proposed strategy can significantly reduce the average link cost of resource access, effectively improve the cache hit rate. The main contributions of this paper are as follows: (1) For the first time, this paper builds a unified model of the multi-level cooperative caching system in mobile CDN scenario. In this model, the core network cache, the RAN cooperative cache and the user-assisted cooperative cache are united to solve the resource delivery and cache replacement problem of cache nodes in the same level and between different levels. (2) According to this model, the problem is closely related to a knapsack problem which is known as NP-Complete problem and hard to solve. Assuming that the optimal solution can be obtained with linear time, many operations such as resource transfer and cache replacement need to be performed between the optimal solutions at different moments, which would result in heavy transmission load and mobile data consumption. Thus a utility-based cooperative caching strategy UCC is proposed to make user access resources as close as possible. (3) Simulation experiments and trace-based verification experiments are designed to verify the performance of the proposed strategy, and other currently widely used caching strategies are implemented in mobile CDN system for comparison. Simulation results show that UCC can reduce the link cost by 10% and improve cache hit rate by 20% compared with the cooperative list-based strategy cLRU(m). Furthermore, we design trace-based verification experiment using real data from China Mobile. The experimental results show that the proposed strategy still has obvious advantages over other widely used caching strategies under the conditions of different load.

Keywords mobile content delivery network; mobile edge computing; hierarchical cooperative caching; cache replacement strategy; device-to-device

1 引 言

随着互联网流量的爆炸式增长,用户对资源访问体验的需求不断提升,内容分发网络(Content Delivery Network, CDN)已成为现代网络基础架构中不可或缺的部分. CDN 大幅度拉近了用户与资源间的距离,保证用户的资源访问体验. 近年来,与 CDN 理念相近的一种新型计算方式——移动边缘计算(Mobile Edge Computing, MEC)得到了学术界和工业界的广泛关注^[1]. CDN 与 MEC 的结合,则催生出一种 CDN 节点下沉的新型解决方案^[2]. 该方案将 CDN 节点部署至移动网络核心网,甚至部署在接入网基站侧^[3],以进一步拉近用户与资源的

距离,减小网络时延,提高用户访问体验. 随着 LTE (Long Term Evolution)的演化和 5G 网络的逐渐成熟,移动网络容量和访问质量不断提升,在移动网络内部部署缓存系统成为可能^[4-5]. 移动 CDN 是在传统 CDN 的基础上将内容节点进一步下沉到 LTE 核心网、接入网侧的一种增强型的内容分发网络. 随着 LTE 网络的大规模部署和应用,大流量、高带宽业务占比不断增加,移动 CDN 将成为进一步提升互联网内容分发效能、更好地满足移动用户的业务访问需求、应对 LTE 网络中流量冲击^[6]的重要手段和发展趋势. 移动 CDN 在网络中的架构如图 1 所示,充分利用了移动网络基础设施,填充了骨干网 CDN 节点与用户之间的空白.

移动 CDN 需要针对移动网络在多方面对传统

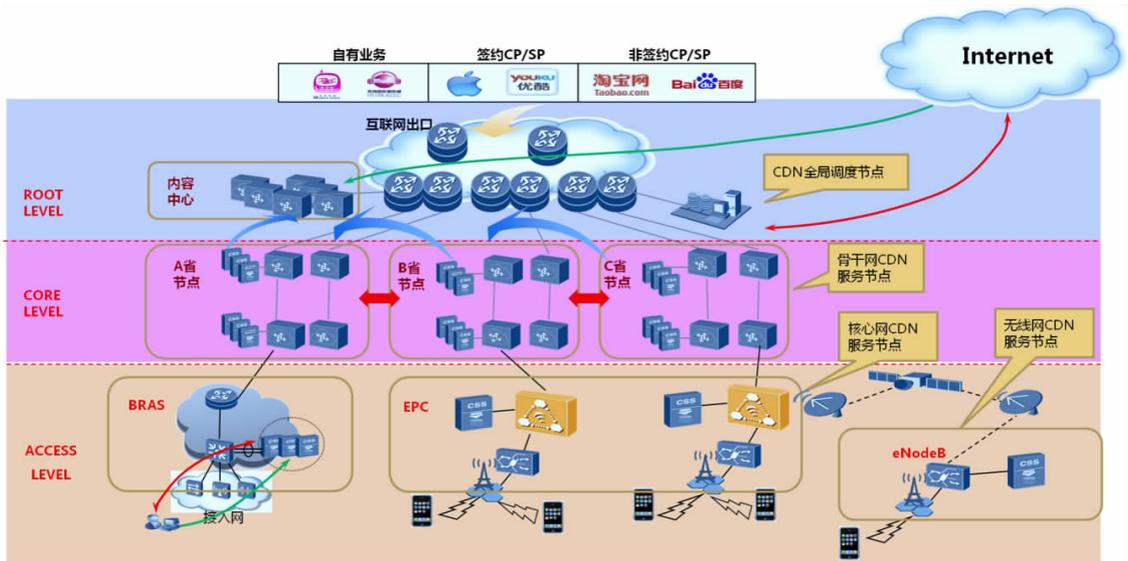


图 1 移动 CDN 延伸骨干网 CDN 与用户之间的最后一公里

CDN 进行改进和优化,包括 LTE 协议适配与兼容性设计、请求调度、移动性支持、计费管理、内容协同调度、业务支撑、负载均衡等。本文着眼于移动 CDN 系统的内容缓存算法和多节点协同策略。移动 CDN 内容节点下沉到核心网后,可以获得移动终端用户的网络信息,从而提供细粒度的控制。接入网侧的内容节点下沉到基站设备上,其主要特点是单节点覆盖用户少、单节点存储容量不大、节点数量众多,因此有必要进行缓存协同。用户终端的介入和用户的移动引入了移动 CDN 缓存内容的动态变化性,同时,用户既是请求产生节点也是缓存内容节点,所以,传统 CDN 的内容分发和缓存策略不直接适用于移动 CDN,需要针对移动 CDN 特定的系统架构,根据用户请求特性,设计动态的内容分发和缓存机制。

如图 2 所示,文献[4]定义了移动 CDN 系统的逻辑结构。多个智能基站(eNodeB)与核心网络处理单元(Evolved Packet Core,EPC)通过回程(backhaul)

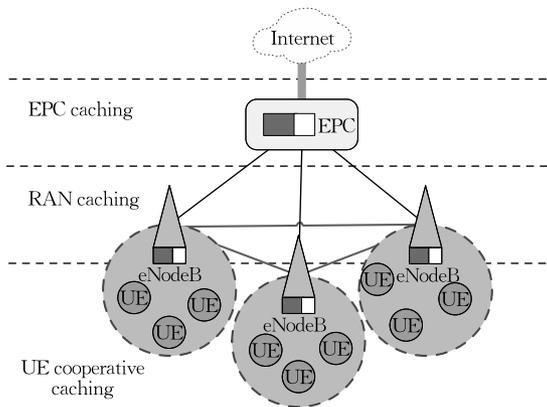


图 2 移动 CDN 分层缓存架构

线路相连;多个 eNodeB 之间通过高速链路互相连接,组成 RAN(Radio Access Network);每个 eNodeB 通过空口无线网络连接一定数量用户设备(User Equipment,UE)。移动 CDN 具备三级缓存:

(1)EPC 缓存。EPC 入网侧部署的缓存服务器。

(2)RAN 缓存。eNodeB 通常具备容量较小的存储空间,可部署缓存服务器。由于多个 eNodeB 之间通过高速链路直连,其缓存内容可共享。

(3)UE 缓存。每个 UE 具备一定缓存空间,利用无线网络的上传/下载链路或者 D2D(Device-to-Device)信道可实现缓存共享。

移动 CDN 架构中,EPC 缓存是单一服务器缓存,而 RAN 层和 UE 层采用协作缓存。协作缓存的目标是增加缓存内容多样性,减小用户平均下载延时,降低网络总体流量。在用户侧进行协同内容分发甚至可以减小移动终端能耗^[7-8]。当前,UE 层协作缓存在技术上可以实施,但存在用户间共享的流量由谁买单的问题。国内运营商现行的流量计费方式不区分用户的上传下载,也不区分互联网访问和基站局部网络内部访问。经测试,同一基站下用户之间传输数据,虽然达到极高传输速率,但上传和下载流量都需要计费。移动 CDN 协作缓存需要 CDN 和通信运营商达成一种商业模式,对用户基站内部流量给予一定优惠减免或补贴,激励用户提供存储资源^[9-10]。此外,用户协作缓存恰符合 D2D 技术的应用场景,D2D 技术的研究与发展也为移动 CDN 的推行带来契机。

移动 CDN 是包含 EPC 单节点缓存、RAN 层协

作缓存和 UE 层协作缓存的分层缓存系统. 在该系统中, 每层缓存该存储哪些资源, 采取何种缓存替换策略, 多层缓存节点之间该如何配合, 才能实现缓存效益的最大化, 这是移动 CDN 实际部署过程中需解决的关键问题, 当前尚无确切解决方案. 本文对该问题进行分析建模, 提出基于效用的协作缓存策略 (Utility-based Cooperative Caching, UCC) 并进行了实验验证. 模拟仿真实验表明, 与基于多级队列的协作缓存策略 cLRU(m) 相比, 本方案能够减小 10% 的链路代价, 缓存命中率提高 20%. 本文在此基础上进一步设计了基于真实网络数据的验证实验, 实验结果表明, 在不同负载情况下, 相比其他目前广泛使用的缓存策略, 所提策略仍然具有明显优势. 本文做出的贡献如下:

(1) 首次针对移动 CDN 系统中多节点分层协同缓存场景进行统一的建模分析, 将核心网缓存、接入网协作缓存以及移动终端侧缓存的资源内容进行关联, 解决同一层次内部、不同层级节点之间的内容协同分发和缓存替换问题.

(2) 提出一种基于全局效用值的协作缓存策略 UCC, 使用户尽可能就近获取资源, 从而提升内容分发性能.

(3) 通过仿真实验和基于真实网络数据的验证实验对所提策略进行验证, 并针对移动 CDN 系统实现了其他目前广泛使用的缓存策略用于对比. 结果表明, 所提策略可以大幅度减小资源访问的平均链路代价, 有效提高缓存命中率.

本文第 2 节综述协作缓存策略相关研究工作; 第 3 节阐述移动 CDN 分层协作缓存模型; 第 4 节描述基于全局效用值的协作缓存策略 UCC; 第 5 节分别通过模拟仿真实验和基于真实数据的验证实验分析所提策略的性能; 第 6 节给出总结和下一步工作.

2 相关工作

移动 CDN 的网络架构是近年来的研究热点^[9, 11-14]. 文献[11]分析了移动 CDN 在 RAN 层的特点, 提出一种分布式的服务架构, 收集大数据并基于数据分析进行网络优化. 文献[12]描述了 5G 系统服务架构, 将移动 CDN 作为一种基础服务. 文献[13]基于信息感知和预测, 提出一种协同代理缓存、预先提取资源的机制. 文献[14]重点描述了移动 CDN 的部署架构, 并针对内容请求调度、缓存管理、负载均衡和无缝切换等方面提出方案. 文献[9]探究了无

线网络场景下采用 D2D 模式进行内容分发的技术.

D2D 技术的研究发展为用户设备之间进行内容共享提供条件. 作为移动网络的延伸, D2D 虽然当前未被部署, 但已进行了诸多研究^[15-18]. 如文献[15]所指出, D2D 允许地理位置相近的设备通过专用信道直接进行数据传输. 基站仅需要检测某一请求符合 D2D 传输条件, 并对其分配专用信道, 起到会话管理功能. D2D 传输可以极大地提高无线频谱资源利用率, 提高传输速率.

移动 CDN 系统包含接入网侧和用户侧两层协作缓存. 协作缓存策略和传统的缓存替换策略不同, 需要考虑全局信息, 以实现全局性能优化. 协作缓存思想最早出现在有线网络中^[19], 随后这一概念被应用在 ad hoc 网络^[20-21]. 文献[22]最早提出移动网络基础设施架构中的 P2P (Peer-to-Peer) 协作缓存策略 COCA (COoperative CAching). COCA 将主机分为高移动性主机和低移动性主机, 后者中存有资源备份. 当用户请求资源时, 先通过 P2P 传输信道广播在邻居节点查找资源, 若查找失败再到基站请求该资源. 基于 COCA 的一系列 P2P 的协作缓存策略被相继提出^[23-25]. GroCoca (Group-based COCA)^[24-25] 将移动主机根据运动性和资源获取兴趣分为不同的集群, 同一集群中共享缓存; 并提出相应的分布式动态邻居发现协议. 上述一系列协作缓存策略, 多考虑节点间信息同步机制, 与移动网络架构关联不密切. 虽然声称在移动网络中使用, 但仍保持 ad hoc 特性: 邻居节点通过广播进行发现和通告, 根据能耗调整传输范围. 这种自组网的传输模式, 不适用于智能手机等移动终端; 其能否在真实的环境中保证传输质量, 减小传输时延, 仍有待验证.

随着互联网内容的急剧增加, 协作缓存策略更多地考虑缓存内容的放置和替换问题^[26-29], 尝试建立统一模型, 进行集中决策. 文献[26]对分布式 CDN 缓存系统缓存放置问题进行建模, 并简化为线性问题分析, 给出启发式策略. 所述分布式 CDN 系统包含单一根节点、父节点以及协作缓存的众多叶子节点, 构成树形结构. 与移动 CDN 包含多层协作缓存系统不同, 该文献只考虑了叶子节点一层协作缓存的情形, 未考虑多层协作缓存之间的配合. 文献[27]提出了 Internet 服务提供商不同自治域之间的协作缓存模型. 该文献说明协作缓存资源最优放置问题为 NPC 问题, 并给出启发式缓存策略. FemtoCaching^[28] 首次在移动网络真实部署了协作缓存系统. 该方案在移动基站下搭建若干 backhaul 带宽小、存储容量大

的微基站 (femto), 存储热门资源, 以提高无线频谱资源利用率. 该文献研究了多微基站服务多用户的系统中微基站的资源放置问题, 并提出启发式缓存策略. 文献[29]探究在移动 CDN 中利用基站进行协作缓存的内容放置问题, 分析其为 NP-hard 问题, 将其转化为在网关处的流量最小化问题, 提出启发式缓存算法. 该文献研究的也是单层协作缓存机制, 与本文所述的移动 CDN 分层协作缓存场景存在明显区别.

缓存替换策略是 CDN 系统的关键技术, LRU (Least Recently Used) 是最基础的策略之一, 由于其简单性和有效性, 基于 LRU 的缓存策略当前仍得到工业界的广泛使用和学术界的广泛关注^[30-33]. 文献[30]提出了基于多级队列的 LRU 策略 $LRU(m)$, 并与基于多级队列的缓存策略族进行了性能对比, 对比策略包括 $FIFO(m)$ 、 $RANDOM(m)$ 等. 结果表明, 基于多级队列的缓存策略性能优于 LRU, 其中 $LRU(m)$ 策略性能最优. 文献[31]将缓存策略根据作用阶段细分为缓存准入策略和缓存替换策略, 并指出大多数缓存策略仅考虑了替换策略, 没有考虑准入策略, 如 LRU. 缓存准入策略即缓存新资源的规则, 文献[32]提出的 BloomFilter 策略是一种典型的准入策略, 该策略使用布隆过滤器 (Bloom Filter) 记录资源请求, 对只有一次请求的资源进行过滤, 当第一次请求某资源时, 仅将该资源记录在 Bloom Filter 中, 不缓存资源; 第二次请求该资源时, 再按 LRU 策略进行缓存和替换, 从而减小磁盘写入次数, 同时避免冷门资源的单次请求造成的缓存污染. 该策略被全球最大 CDN 服务商 Akamai 所采用. 文献[33]提出了 LRU 策略的协作缓存版本 cLRU (cooperative LRU). 当本地没有缓存某资源, 若进行协作缓存的其他节点缓存了该资源, 则从其他节点获取该资源, 但自身节点不缓存该资源. 该策略保证协作缓存空间每个资源有一个副本, 最大化缓存空间利用率. 传统缓存策略在移动 CDN 系统中的性能有待验证.

综上所述, 现有的缓存策略与移动 CDN 分层缓存系统不完全契合. 基于 ad hoc 的协作缓存策略不具备实施的现实条件. 集中式协作缓存的相关研究利用了移动网络集中化特性, 针对同一层次的协作缓存、以及有父节点的协作缓存提出解决方案, 基于全局优化目标建立统一模型, 并基于贪心策略提出启发式缓存策略. 该类模型未能解决移动 CDN

系统中分层协作缓存的资源协调放置问题. 传统缓存策略及其协作缓存版本可进行进一步修改, 使之适用于移动 CDN 分层协作缓存系统, 但其缓存性能需要验证. 本文针对移动 CDN 分层协作缓存系统进行建模分析, 设计相应的协作缓存策略, 并验证了传统缓存策略的性能.

3 移动 CDN 分层协作缓存模型

本节对移动 CDN 分层协作缓存系统建立统一的最优化模型.

如图 3 所示, 在移动 CDN 分层协作缓存系统中, 当 UE 请求某一资源时, 其可能的来源有: 自身缓存、所属 eNodeB 缓存、其他 eNodeB 缓存、其他 UE 缓存、EPC 缓存和 Internet. 假设用户的请求兴趣已经给出, 资源查找和请求收发已采取恰当方式进行. 在实际系统中, 用户兴趣一般通过历史请求数据分析或在线实时方式^[34] 获得, 资源查找和传输通常采用缓存交换协议^[35,36] 和 GPRS 隧道技术 (GTP).

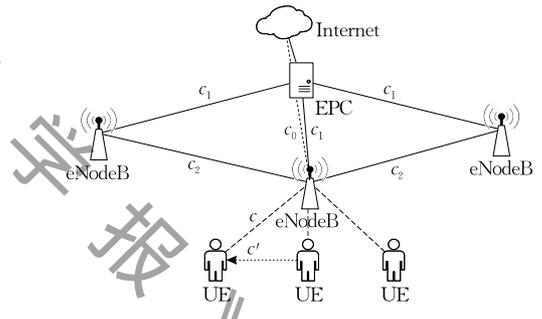


图 3 链路代价示意图

模型使用链路代价表示网络中端到端连接的网络传输成本. 链路代价不是实际物理量, 它综合了时延和带宽, 反映了端到端数据传输的网络代价, 可以通过测量时延、带宽等估算链路质量. 模型中我们假设链路质量是已知的. 链路代价越大, 表示该链路时延高、带宽低、稳定性差. 假设 eNodeB 与 Internet 之间的链路代价为 c_0 , eNodeB 与 EPC 之间 backhaul 链路代价为 c_1 , eNodeB 之间的链路代价为 c_2 , UE 与其所在 eNodeB 之间的空口链路代价为 c , UE 与其他 UE 之间无线链路代价为 c' . 在实际的网络中, 不同 eNodeB 之间的传输成本、eNodeB 和用户之间的传输成本是不同的, 但不影响所述模型和算法效果. 所以简便起见, 我们将链路代价设置为常数, 并在 5.1.3 节实验部分对传输成本不同的情况进行了讨论和验证. 链路代价如图 3 和表 1 所示.

表 1 链路代价

来源	链路代价
自身缓存	0
所属 eNodeB	c
协作 eNodeB	$c_2 + c$
协作 UE	c'
EPC 缓存	$c_1 + c$
Internet	$c_0 + c$

由于从 eNodeB 到 Internet 的链路经过 RAN 和 EPC 之间的 backhaul 链路和 Internet 内部,故 $c_0 > c_1$. 由于 eNodeB 之间通过高速链路连接^[4],而 RAN 和 EPC 之间的 backhaul 链路仍存在较大传输时延,故 $c_0 > c_1 > c_2$; 由于 UE 之间采用无线信道传输,无论是当前通过基站辅助的上行和下行链路传输,还是将来通过 D2D 技术建立会话分配专用信道,其链路代价应高于有线高速链路,故 $c' > c_2$. 当前,UE 之间数据传输采用基站辅助的上传和下载链路,若 UE 从属于同一基站,则 $c' \approx 2c$. 此时,由于 c 为无线链路代价, c_2 为高速的有线链路代价, $c > c_2$, 故 $c' > c_2 + c$. 若 UE 不属于同一基站,则 $c' \approx 2c + c_2$, 显然, $c' > c_2 + c$. 未来,UE 之间的数据传输采用 D2D 模式,对于地理位置较为接近的两个 UE, 基站只负责建立 D2D 会话,会话建立后,通过 D2D 专用信道传输,可实现较高传输带宽. 此时,可能 $c' < c_2 + c$. 模型中,假定 $c' > c_2 + c$, $c' < c_1 + c$.

假设有 K 个 eNodeB, M 个 UE. 有 N 个资源文件,每个文件大小为 s_n . 每个 UE 的缓存空间大小为 B_m , 每个 eNodeB 的缓存空间大小为 Q_k , EPC 处缓存大小为 V . 假设用户对资源文件的请求率为 d_m^n . d_m^n 表示一段时间内用户 UE_m 请求某个资源 n 的概率,它反映了用户对某个资源的兴趣度.

如表 2 所示,使用 0-1 变量 x_m^n 表示 UE_m 是否缓

表 2 模型参数及变量

变量	含义
K	eNodeB 集合,集合中元素个数为 K
M	UE 集合,集合中元素个数为 M
N	资源文件集合,集合中元素个数为 N
s_n	资源 n 的大小
B_m	UE_m 的缓存空间
Q_k	$eNodeB_k$ 的缓存空间
V	EPC 的缓存空间
d_m^n	UE_m 对资源 n 的请求率
x_m^n	缓存决策 0-1 变量, UE_m 是否缓存资源 n
y_k^n	缓存决策 0-1 变量, $eNodeB_k$ 是否缓存了资源 n
z^n	缓存决策 0-1 变量, EPC 是否缓存了资源 n
X_{mi}^n	访问控制 0-1 变量, UE_m 是否从 UE_m 获取资源 n
Y_{mk}^n	访问控制 0-1 变量, UE_m 是否从 $eNodeB_k$ 获取资源 n
Z_m^n	访问控制 0-1 变量, UE_m 是否从 EPC 获取资源 n

存了资源 n , 使用 0-1 变量 y_k^n 表示 $eNodeB_k$ 是否缓存了资源 n , 使用 0-1 变量 z^n 表示 EPC 缓存是否缓存了资源 n . 使用访问控制辅助 0-1 变量 X_{mi}^n , Y_{mk}^n 和 Z_m^n 表示 UE_m 获取资源 n 的来源, 具体为 X_{mi}^n 表示 UE_m 是否从 UE_i 获取资源 n , Y_{mk}^n 表示 UE_m 是否从 $eNodeB_k$ 获取资源 n , Z_m^n 表示 UE_m 是否从 EPC 缓存获取资源 n .

移动 CDN 分层协同缓存问题可描述为: 针对包含两层协作缓存平面的网络拓扑, 给定一定数量资源和一定数量用户, 如何协同层级内部、不同层级之间节点的内容放置, 如何确定各个节点上的资源缓存策略, 并为每个用户的资源请求指定资源下载源, 以最小化所有用户获取资源的总代价. 此问题实际分为两部分: 缓存策略和资源获取来源决策. 前者是节点在有限的缓存空间限制下该缓存哪些资源, 后者是为资源访问请求选择源节点. 在考虑获取资源的链路代价时, 二者相互关联, 相互影响. 通常情况下, 用户希望获取资源的链路代价最小. 故获取来源决策可采取从链路代价最小的来源获取的简单策略. 模型中, 默认采取此来源决策策略, 从而考虑缓存策略.

因此, 模型的变量为 x_m^n , y_k^n 和 z^n . 访问控制变量 X_{mi}^n , Y_{mk}^n 和 Z_m^n 可以根据 x_m^n , y_k^n 和 z^n 推导得出. 即在链路访问代价大小已知的情况下, 在对任意资源 n , 只要给定 x_m^n , y_k^n 和 z^n , 对任意 UE_m ($m \in M$), 均可直接得出 X_{mi}^n , Y_{mk}^n 和 Z_m^n . 在移动 CDN 系统中, 资源获取来源优先级顺序为 UE 自身缓存, 所属 eNodeB, 协作 eNodeB, 协作 UE, EPC 缓存, Internet, 即 $c < c_2 + c < c' < c_1 + c < c_0 + c$. 在图 4 所示系统中, 有 1 处 EPC 缓存, 3 处 eNodeB 缓存 ($eNodeB_k$, $k = 1, 2, 3$), 每个基站下各有 1 个 UE (UE_m , $m = 1, 2, 3$). 设有资源 a , 缓存位置为 EPC, $eNodeB_2$ 和 UE_1 . 即 $z^a = y_2^a = x_1^a = 1$, $y_1^a = y_3^a = x_2^a = x_3^a = 0$. 此时对于 UE_m ($m = 1, 2, 3$), 其访问决策变量 X_{mi}^a , Y_{mk}^a 和 Z_m^a 各

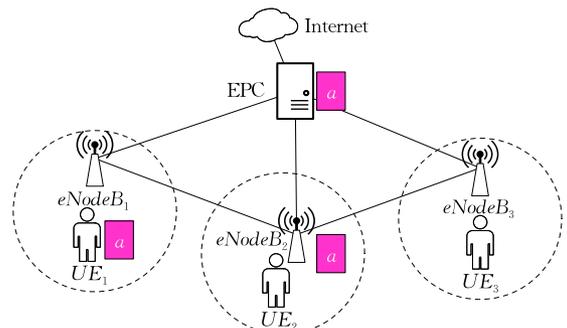


图 4 资源获取示意图

为 $X_{1i}^a = Y_{1k}^a = Z_1^a = 0$, 表示 UE_1 从其自身缓存获取资源 a ; $X_{2i}^a = Y_{21}^a = Y_{23}^a = Z_2^a = 0, Y_{22}^a = 1$, 表示 UE_2 从 $eNodeB_2$ 获取资源 a ; $X_{3i}^a = Y_{31}^a = Y_{33}^a = Z_3^a = 0, Y_{32}^a = 1$, 表示 UE_3 从 $eNodeB_2$ 获取资源 a . 虽然 UE_2 和 UE_3 同从 $eNodeB_2$ 获取资源 a , 但它们的链路代价不相同.

移动 CDN 分层协作缓存模型的优化目标为总体链路代价, 使所有用户获取资源的链路代价尽可能小, 也就是说, 与全都从 Internet 获取资源相比, 使用户获取资源节省的链路代价尽可能大. 定义全局效用值 *Utility*, 表示资源缓存节约的链路总代价. 效用机制是互联网体系结构评估的常用方法^[37-38], 广泛应用于资源分配^[39]和流量工程^[40].

$$\max. \text{Utility} = \sum_{m \in M} \sum_{n \in N} s_n \cdot d_m^n \left((c_0 + c) x_m^n + c_0 \cdot Y_{m, k(m)}^n + (c_0 - c_1) Z_m^n + \sum_{k \in K, k \neq k(m)} (c_0 - c_2) Y_{mk}^n + \sum_{i \in M, i \neq m} c \cdot X_{mi}^n \right) \quad (1)$$

sub.

$$\sum_{n \in N} s_n \cdot x_m^n \leq B_m, \quad \forall m \in M \quad (2)$$

$$\sum_{n \in N} s_n \cdot y_k^n \leq Q_k, \quad \forall k \in K \quad (3)$$

$$\sum_{n \in N} s_n \cdot z^n \leq V \quad (4)$$

$$X_{mi}^n \leq x_i^n, \quad \forall m \in M, \forall i \in M \text{ 且 } m \neq i, \forall n \in N \quad (5)$$

$$Y_{mk}^n \leq y_k^n, \quad \forall m \in M, \forall k \in K, \forall n \in N \quad (6)$$

$$Z_m^n \leq z^n, \quad \forall m \in M, \forall n \in N \quad (7)$$

$$x_m^n + \sum_{k \in K} Y_{mk}^n + Z_m^n + \sum_{i \in M, i \neq m} X_{mi}^n \leq 1, \quad \forall m \in M \quad (8)$$

式(1)中, $k(m)$ 表示 UE_m 所属的 $eNodeB$, M_m^* 表示 UE_m 所属基站范围内的 UE 集合.

在非 D2D 模式下, 即 UE 通过基站的上传和下载链路互相访问, 当 UE 处于同一基站下 $c' \approx 2c$; 当 UE 不属于同一基站, $c' \approx 2c + c_2$. 此时, $\sum_{i \in M, i \neq m} c^*$ 可表示为

$$\sum_{i \in M, i \neq m} c^* = \sum_{i \in M_m^*} (c_0 - c) + \sum_{i \in M \setminus M_m^*} (c_0 - c_2 - c) \quad (9)$$

D2D 模式下, 如上文所指出, c' 的大小不能确定, 故 $c^* = c_0 + c - c'$.

优化目标中的 $(c_0 + c) x_m^n$ 表示如果从自身缓存获取资源, 相比于直接从 Internet 获取资源, 节约的链路代价为 $c_0 + c$; $c_0 \cdot Y_{m, k(m)}^n$ 表示如果从所属基站获取资源, 节约的链路代价为 c_0 ; $(c_0 - c_1) Z_m^n$ 表示如果从 EPC 缓存获取资源, 节约的链路代价为

$c_0 - c_1$; $\sum_{k \in K, k \neq k(m)} (c_0 - c_2) Y_{mk}^n$ 表示如果从其他协作基站获取资源, 节约的链路代价为 $c_0 - c_2$; $\sum_{i \in M, i \neq m} c^* \cdot X_{mi}^n$ 表示如果从同一基站下的其他 UE 获取资源, 节约的链路代价为 c^* .

在约束条件中, 式(2)~(4)分别是 UE、eNodeB 和 EPC 的缓存空间约束; 式(5)~(7)的含义是, 只有当获取资源的源本身缓存了该资源, 才能为 UE 提供该资源; 式(8)约束的含义是, 在计算节约的链路代价时, UE 要么从自身获取资源, 要么从 eNodeB 获取资源, 要么从 EPC 获取资源, 要么从其他 UE 获取资源, 要么从 Internet 获取资源. 优化目标中的链路代价参数与移动 CDN 系统中的不同层级一一对应, 它们的因子决策变量则决定了用户从哪个节点获取资源. 式(8)的约束体现了移动 CDN 系统中多级协作缓存的特点. 它隐含了缓存决策的优先级, 即为保证优化目标向最优解逼近, 用户需从链路代价小的节点获取资源. 对于 RAN 层的协作缓存, 由于 eNodeB 本身不请求资源, 对其全局效用值的衡量最终转化到用户资源请求的链路代价上. 如此便将 RAN 层和 UE 层两个协作缓存平面统一起来, 可以设计统一的缓存策略, 解决不同协作缓存平面的节点间的内容协同调度问题.

式(8)的约束条件与 P2P 系统不同. P2P 系统中, 允许用户同时从多个邻居节点下载数据, 而式(8)对缓存资源来源进行约束时, 限制用户每次请求从一处获取资源. 在当前尚未支持 D2D 的移动 CDN 系统中, 由于从 eNodeB 缓存处获取资源的链路代价优于从 UE 获取资源, 所以当可以从 eNodeB 获取资源时, 再同时从 UE 获取, 链路代价没有减少. 对于从协作 UE 获取资源的情况, 类似文献[28], 假设每个资源文件是经过分块的, 一个节点对一个资源分块只可以缓存或者不缓存, 此时可将资源看作元数据. 所提模型的优化目标是节约的链路代价, 对于从一个 UE 或者同时从多个 UE 获取资源的情况, 节约的链路代价是相同的. 在真实环境下, 允许存在从多个 UE 获取资源的情况, 这与模型并没有冲突.

在上述模型中, 当从协作 UE 获取资源时, 没有限定协作 UE 的范围. 现实中, 同一基站下的不同 UE 具备相同的子网地址, 能够互相访问, 而 UE 对所属基站之外的网络设备通常情况下是不可见的. 此外, 对于非 D2D 传输模式, 同一基站下的 UE 相互传输, 可近似看成同一无线局域网内通信, 可实现较高传输速率; 不同基站下的 UE 通信, 需要经过基

站间通信链路,如式(1)的右半部分, $c_0 - c_2 - c$ 不一定恒为正.在 D2D 传输模式下,UE 之间需要存在地理位置的接近关系,而不同基站下的 UE 距离较远.所以,针对这种情况,我们限制 UE 之间的通信只能在同一基站下进行.不失一般性,对模型做如下约束:

$$\sum_{i \in M \setminus M_m^*} x_{mi}^n = 0, \forall n \in N, \forall m \in M \quad (10)$$

这一约束缩小了解空间,降低了模型难度.

模型中,变量 x_m^n, y_k^n, z^n 均为 0-1 变量,表示该节点是否缓存了某资源.可将缓存每个资源所节约的链路代价看作其价值,模型约束条件 2 可看作容量约束.由此,所提模型可看作一类背包问题.由于背包问题属于 NPC 问题,故移动 CDN 分层协作缓存模型也属于 NPC 问题. NPC 问题线性时间内无法求得最优解.此外,对于移动 CDN 缓存系统而言,缓存状态根据用户的实际请求而变化.假设能够线性求得每个时刻的最优解,即每个时刻各个内容节点的最优内容放置.那么,不同时刻之间最优内容放置不同,需要发生大量的资源传输和替换等操作,造成网络开销和用户流量的消耗,而此时用户不一定请求资源.所以,从用户实际请求出发,采用启发式策略动态地进行资源更新是代价最小的可行方案.因此,针对移动 CDN 分层协作缓存模型的内容放置和替换问题,我们设计了启发式算法.

4 基于全局效用值的协作缓存策略

由于移动 CDN 分层协作缓存问题是 NPC 问题,同时考虑不同缓存状态之间切换的代价,我们从用户请求出发,基于贪心思想设计了启发式算法.该算法的核心在于,在进行缓存决策的每一步,保证全局优化目标值的增加.

算法 1. 资源获取来源 $get_source(i, n)$.

输入:发起请求的用户 i ,请求的资源 n

输出:资源来源层次 $level$, 资源来源编号 $source$

1. $LEVEL = \{INTERNET, EPC, A_ENODEB, ENODEB, A_UE, UE\}$
2. IF n in i THEN
3. RETURN (UE, i)
4. ELSE IF n in $eNodeB_i$ THEN
5. RETURN $(ENODEB, eNodeB_i)$
6. END IF
7. FOR $eNodeB$ in K DO
8. IF n in $eNodeB$ THEN

9. RETURN $(A_ENODEB, eNodeB)$
10. END IF
11. END FOR
12. $M = \{ue \text{ for } eNodeB_{ue} = eNodeB_i\}$
13. FOR ue in M DO
14. IF n in ue THEN
15. RETURN (A_UE, ue)
16. END IF
17. END FOR
18. IF n in epc THEN
19. RETURN (EPC, epc)
20. END IF
21. RETURN $(INTERNET, null)$

由于在资源获取来源决策问题上采用最小化链路代价的策略,在缓存决策问题上,需要确定资源获取来源优先级顺序.如上文所述,根据模型假设,按链路代价从小到大排列,其来源优先级大小关系为所属 $eNodeB >$ 协作 $eNodeB >$ 协作 $UE >$ $EPC >$ $Internet$.在计算某一资源获取的链路代价时,只要给定一种资源放置策略,就能根据此优先级得到资源获取缓存决策,从而得到相应链路代价.算法 1 描述了根据资源放置确定资源获取来源的方法.

如式(11),计算缓存某一资源后所有 UE 对该资源的平均访问链路代价,减去缓存这一资源前所有 UE 对该资源的平均访问链路代价,即缓存这一资源造成的平均访问代价的减小值,作为缓存决策因子.我们定义 $\Delta Utility_i^n$,表示节点 i 缓存资源 n 后,所有 UE 获取资源 n 的总链路代价减小值,也就是所有 UE 获取资源 n 总节约的链路代价增加值.

$$\begin{aligned} \Delta Utility_i^n = & \sum_{m \in M} d_m^n \cdot ((c_0 + c)(x_m'^n - x_m^n) + c_0(Y_{m \cdot k(m)}'^n - Y_{m \cdot k(m)}^n) + \\ & (c_0 - c_1)(Z_m'^n - Z_m^n) + \\ & \sum_{k \in K, k \neq k(m)} (c_0 - c_2)(Y_{mk}'^n - Y_{mk}^n) + \\ & \sum_{i \in M, i \neq m} c^* \cdot (X_{mi}'^n - X_{mi}^n)) \end{aligned} \quad (11)$$

其中, $(x_m'^n, Y_{m \cdot k(m)}'^n, Z_m'^n, Y_{mk}'^n, X_{mi}'^n)$ 是节点缓存资源 n 后,各 UE 获取资源 n 的访问决策变量, $(x_m^n, Y_{m \cdot k(m)}^n, Z_m^n, Y_{mk}^n, X_{mi}^n)$ 是节点缓存资源 n 之前,各 UE 获取资源 n 的访问决策变量.

当缓存容量充足时,直接计算 $\Delta Utility_i^n$ 值,缓存该资源;当缓存空间不足时,需要进行缓存替换.替换策略为,若资源文件的 $\Delta Utility_i^n$ 值小于缓存中所有文件的 $\Delta Utility_i^n$ 值,则不缓存;否则,选择 $\Delta Utility_i^n$ 值最小的文件替换,保证在缓存空间能容

纳的情况下全局 $Utility$ 值的增加. 如果同一节点的两个资源 $\Delta Utility_i^n$ 值相同, 则对该资源采用 LRU 策略. 算法 2 描述了 UCC 策略的核心处理过程.

算法 2. UCC 策略流程.

输入: 发起请求的用户 i , 请求的资源 n

输出: \emptyset

1. $LEVEL = \{INTERNET, EPC, A_ENODEB, ENODEB, A_UE, UE\}$
2. $(level, source) = get_source(i, n)$
3. $pos = []$
4. IF $level == INTERNET$ THEN
5. $pos.add(epc)$
6. END IF
7. IF $level < ENODEB$ THEN
8. $pos.add(eNodeB_i)$
9. END IF
10. IF $level < UE$ THEN
11. $pos.add(i)$
12. FOR p in pos DO
13. $ut = calculate_utility(p, n)$
14. IF $volume_{left} \geq s_n$ THEN
15. Cache resource n
16. ELSE IF $sum(ut_{res_list}) < ut$ AND $sum(size_{res_list}) \geq s_n$ THEN
17. Cache resource n , replace resource res_list
18. ELSE
19. Do not cache resource n
20. END IF
21. Update utility of n and res_list in other nodes
22. END FOR

由于 EPC 缓存为单节点缓存, 可以计算全局优化目标增益 $\Delta Utility_i^n$, 作为缓存决策因子. 也可以将 EPC 缓存与 RAN 和 UE 协作缓存分开考虑, 直接采用 LRU 或 LFU 缓存策略. 这种情况下, EPC 缓存服务器的作用相当于将部分 CDN 服务器部署于运营商网络内部. 我们采用两种策略的结合, 即首先根据 $\Delta Utility_i^n$ 进行缓存替换, 当存在两个资源 $\Delta Utility_i^n$ 相同时, 进而使用 LRU 策略区分.

在移动 CDN 分层协作缓存系统中, 需要一个全局决策节点进行缓存调度, 该节点可以部署在 EPC 处. 每当请求发起, 决策节点接收该请求, 对该请求进行服务, 并进行缓存决策. 决策节点维护所有基站和用户的缓存信息, 对于资源分块存储的情况, 其缓存信息如下:

(1) 当前 $\Delta Utility_i^n$ 最小的资源编号及其全局效用值: $(n_{min}, \Delta Utility_i^{min})$.

(2) 当前缓存的资源列表: N_i^* .

进行缓存决策时, 发生缓存替换的节点只需要比对新缓存的文件 k 的 $\Delta Utility_i^k$, 如果 $\Delta Utility_i^k > \Delta Utility_i^{min}$, 则替换资源 n_{min} , 重新计算 $(n_{min}, \Delta Utility_i^{min})$; 更新其缓存资源列表. 对于其他节点而言, 如果缓存替换节点缓存和替换的资源分别是 p 和 q , 则需要更新缓存了资源 p 或资源 q 的节点的缓存信息: 例如, 若节点 j 缓存了资源 p 或 q , 只需重新计算 $\Delta Utility_j^p$ 或 $\Delta Utility_j^q$, 据此更新 $(n_{min}, \Delta Utility_i^{min})$ 即可. 由于存在 UE 突然离开或加入协作缓存的情况, 需要周期性地对缓存的全部资源的 $\Delta Utility$ 值进行更新. 对于资源未经分块的情况, 需按照 $\Delta Utility_i^n$ 值有序存储, 进行缓存替换时, 由于资源容量限制, 可能替换多个资源.

资源缓存有预置资源(pre-fetched)和非预置资源(no-pre-fetched)两种方式. 预置资源指, 根据用户的兴趣预测将来可能被访问的内容, 将这些内容提前缓存在节点中, 以提高缓存命中率. 对于 UE 缓存, 预置资源需要消耗用户流量, 不符合实际应用场景; 对于 eNodeB 缓存, 可以采取预置资源方式, 但预置资源需要准确的资源流行度预测. 如果预置的资源将来一段时间没有被访问, 非但缓存效率不能提高, 还造成额外开销. 非预置资源策略指, 缓存过程只在实际请求发起时发生. 这一策略是符合常理的, 不会造成额外开销. 因此, 所提策略针对非预置资源情形.

对于缓存决策的触发时机, 所提策略采取的方法为由外到内, 缓存决策应在资源获取途径节点上进行. 对于 UE 协作缓存, 获取的资源文件最终要传输至 UE. 故只要有请求, 发起请求的 UE 就应执行缓存决策. 对于请求发起的用户所属的基站, 只有资源来源为互联网、EPC 或其他基站时, 需要进行决策. 当用户从协作缓存的其他用户获取资源时, 虽然由基站进行路由, 但由于基站缓存容量较小, 基站及其对应用户构成协作缓存空间, 所以不缓存该资源. 对于 EPC, 当请求由外部互联网响应时, EPC 需要进行缓存决策. 所提策略与某个资源的存放位置和副本数量无明显相关性, 而是通过计算缓存该资源是否提升全局效用值进行决策.

5 实验评估

本节分别通过模拟仿真实验和基于真实数据的验证实验, 分析了所提策略在平均链路代价和缓存

命中率方面的性能. 针对移动 CDN 分层协作缓存系统, 我们扩展了 cLRU 策略^[33]、BloomFilter 策略^[32], 并在 cLRU 策略的基础上融合了 LRU(m) 策略^[30] (我们称之为 cLRU(m) 策略) 用于对比. 在 cLRU 策略中, EPC 使用 LRU 缓存策略; eNodeB 本身的缓存使用 LRU 缓存策略, 当本地缓存查找失败, 只要其他协作的 eNodeB 缓存了该资源, 该 eNodeB 本身就不缓存该资源; UE 本身的缓存使用 LRU 缓存策略, 当本地查找失败时, 只要所属 eNodeB 或其他协作 UE 缓存了该资源, 该 UE 本身就不缓存该资源. BloomFilter 策略与 cLRU 策略的区别在于缓存准入策略: 在移动 CDN 系统中, UE 使用 LRU 策略; eNodeB 使用布隆过滤器记录该 eNodeB 下的 UE 首次请求的资源, 当第二次请求时再缓存; EPC 同样使用布隆过滤器进行请求过滤. cLRU(m) 策略在 cLRU 的基础上使用多级缓存队列, 理论上具备更优的缓存性能.

5.1 模拟仿真实验分析

模拟仿真中, 我们假设所有资源大小固定, 使用缓存的资源个数表示缓存容量. 设置一个 EPC, 缓存大小 $V=100$; eNodeB 的个数 $K=5$, 缓存大小 $Q=20$; UE 个数 $M=50$, 缓存大小 $B=10$, 每个 eNodeB 下设置 10 个 UE; 资源总数 $N=1000$. 设置链路代价为固定值: $c=0.1$, $c_0=0.6$, $c_1=0.3$, $c_2=0.1$, $c'=0.3$. 使用泊松过程模拟用户请求, 参数 $\lambda=50$. 用户的资源访问热度排名为随机生成, 用户访问请求率由该资源热度排名根据齐夫分布 Zipf(α, q) 产生:

$$R_{jn} = \frac{1}{(r_{jn} + q)^\alpha} \quad (12)$$

齐夫分布是较符合缓存系统中用户实际资源请求的分布^[41], 被广泛用于缓存算法模拟^[26-27, 42]. 上式中, r_{jn} 为资源 n 对节点 j 的资源热度排名, 参数 $q=0$, $\alpha=0.8$, 这一取值与文献[43]中的测量结果相符合, 下文对该参数的影响进行了讨论. 对于 cLRU(m) 策略, 根据文献[30], 我们设置队列个数 $m=3$, 初始队列容量为总容量的 60%.

在仿真模拟中, 我们分别从平均链路代价、缓存命中率、对称与非对称请求、齐夫分布参数变化以及缓存容量变化等几个方面对协作缓存策略的性能进行讨论.

5.1.1 平均链路代价

图 5 作出了使用不同缓存策略时平均链路代价随时间的变化曲线. 初始时刻, 由于缓存系统中没有资源, 所有请求的资源均需要从外部 Internet 获取,

此时链路代价为最大值; 随着请求数的增加, 越来越多的资源被缓存到移动 CDN 分层缓存系统中, 此时大量资源可从缓存中获取, 链路代价减小; 当缓存空间已满, 需要进行缓存替换, 此时平均链路代价称缓慢下降趋势, 最终趋于稳定. cLRU 策略的最终平均链路代价为 0.56, UCC 策略的最终链路代价为 0.44, BloomFilter 策略的最终链路代价为 0.57, cLRU(m) 策略的最终链路代价为 0.54, 相比于 cLRU(m) 策略, UCC 策略平均链路代价降低了 10%.

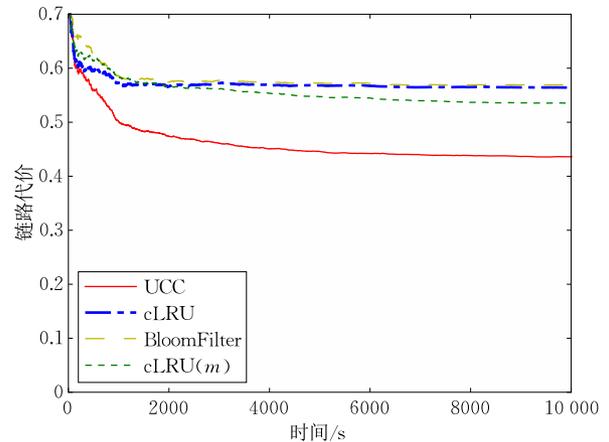


图 5 平均链路代价变化图

5.1.2 缓存命中率

图 6 作出了上述缓存策略中由协作缓存空间满足的请求数变化, 以解释平均链路代价下降的原因. 这里的协作缓存空间指的是使用协作缓存策略的 UE 层和 RAN 层. 由图可以看出, 在全部请求数呈线性增长的情况下, 初始阶段, 不同缓存策略由协作缓存空间满足的请求数基本相同, 随着缓存替换的发生, UCC 策略由协作缓存空间满足的请求数明显高于其他策略.

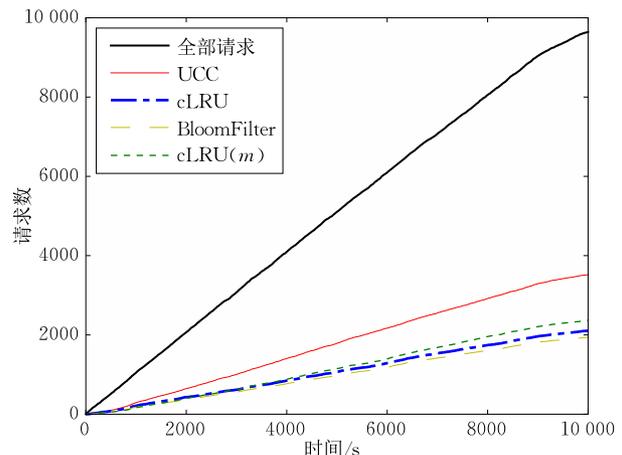


图 6 请求数变化图

图 7 作出了上述策略缓存命中率的变化曲线, 此处缓存命中率指的是包括自身节点和移动网络缓存节点(UE、RAN 以及 EPC)的总体命中率. 图示模拟中, cLRU 策略的缓存命中率最终为 28%, UCC 策略协作缓存命中率最终为 55%, BloomFilter 策略的缓存命中率最终为 28%, cLRU(m) 策略的缓存命中率最终为 35%. 可以看出, BloomFilter 策略与 cLRU 策略性能相当, cLRU(m) 策略的性能优于 cLRU 策略, UCC 策略性能最优. 相比于次之的 cLRU(m) 策略, UCC 策略的缓存命中率提高了 20%. UCC 策略中, 更多的请求命中缓存节点, 从而降低了资源访问的平均链路代价, 提升用户体验.

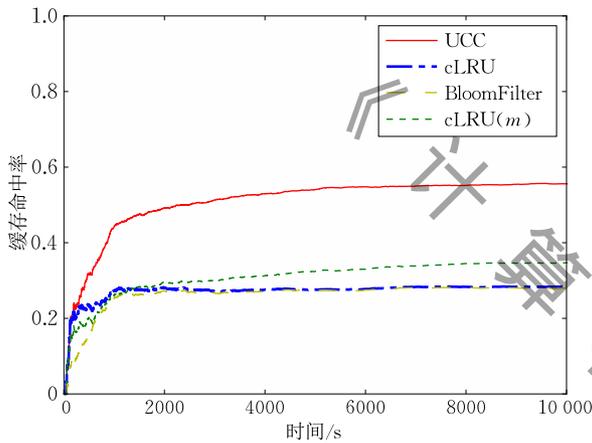


图 7 缓存命中率变化图

5.1.3 传输成本不同时的验证

上述模拟中我们将不同用户、不同基站之间的链路代价设置为常数, 这种假设遵循一种基本关系, 即 eNodeB 之间的高速有线链路优于无线链路, 且优于 backhaul 链路. 为了验证传输成本不同时的算法性能, 设置不同 UE 和 eNodeB 之间、eNodeB 之间的链路代价随机取值, 但仍然基本遵循基本上述基本前提. 具体取值如下: UE 和 eNodeB 之间取 0.1~0.2 的随机链路代价, eNodeB 之间取 0.05~0.15 的随机链路代价, EPC 到 eNodeB 之间仍取 0.3, Internet 上的 CDN 服务器和 eNodeB 之间仍取 0.4. 我们进行了 5 组随机取值, 对不同缓存策略的平均链路代价和缓存命中率进行对比.

对比图 8 和图 9 可以看出, 在链路代价进行随机设置时, UCC 策略在平均链路代价和缓存命中率上的性能均优于其他缓存策略, 且不同组别之间的性能差别不大. 也就是说, 实际网络传输成本不同并不影响所述模型和算法性能. 由于链路代价设置的具体值不同, 不同组别的用户请求的平均链路代价

在数值上有差别. 从图 9 统计的缓存命中率来看, cLRU 策略、BloomFilter 策略和 cLRU(m) 策略的缓存命中率不受链路代价的影响, 这是因为这些策略在进行缓存决策时不考虑链路代价. 而对于 UCC 策略而言, 链路代价影响缓存资源的 *Utility* 值, 所以缓存替换的结果会有差别. 但由于链路代价遵循上文所述有线链路优于无线链路的基本关系, UCC 策略的缓存决策结果受链路代价影响不大, 其性能仍明显优于其它策略.

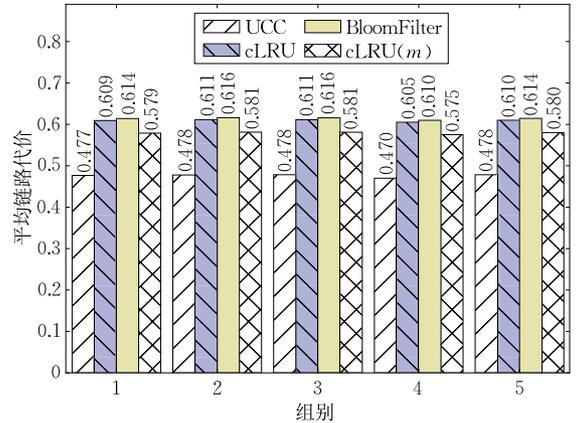


图 8 传输成本不同时的平均链路代价

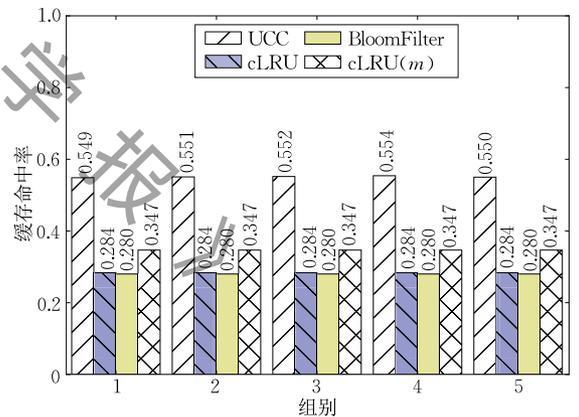


图 9 传输成本不同时的缓存命中率

5.1.4 对称请求与非对称请求的对比

上述模拟中对每个用户随机设置一种资源热度排名, 即用户的兴趣是随机非对称的, 这种设置与通常情况相符. 但考虑可能存在地理位置相近的用户资源兴趣相似的情况, 对每个基站设置一种资源热度排名, 即请求是对称的, 比较缓存命中率和链路代价平均值.

对比图 10 和图 11 中的对称请求和非对称请求的平均链路代价和协作缓存命中率, 可以发现无论使用何种策略, 缓存性能都会明显提升. 这是因为所述策略均为协作缓存策略: 对 UCC 策略而言, 它考

虑的是缓存每个资源的全局效用,当用户请求兴趣趋向一致时,热点资源的总体请求率高,根据式(11),其全局效用就高,故其在协作缓存空间的命中率就高;对其他策略而言,由于在缓存更新时基于最近使用的原则,所以当热点资源的请求频率较高时,其留在缓存空间的可能性就会提高.同时由于资源请求分布大致相同,协作缓存节点可以缓存更多不同资源,缓存空间的利用率将会提高.可以看出,无论是对称请求还是非对称请求,UCC 策略的缓存性能均明显高于其他策略.

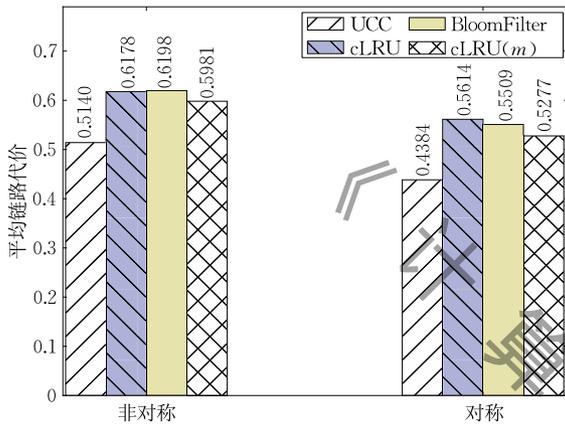


图 10 对称与非对称请求的链路代价

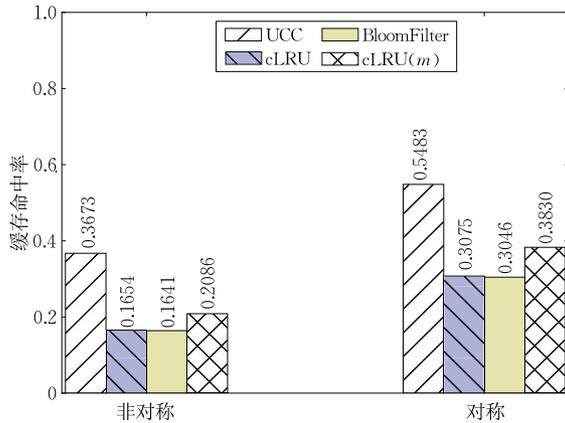


图 11 对称与非对称请求的缓存命中率

5.1.5 齐夫分布参数变化的影响

图 12 和图 13 分别作出了齐夫分布参数 α 取 0.7、0.8、0.9、1.1、1.2、1.3 时的平均链路代价和协作缓存命中率的曲线.参数 α 反映了请求的密集程度, α 越大,请求热点资源的概率越高.由图可以看出,当齐夫分布的参数 α 增大时,各缓存策略的平均链路代价明显减小,缓存命中率明显提升.这是由于请求分布向热门资源集中,局部资源的访问频率提高,该部分资源在缓存空间内更多命中,缓存性能从

而提升.参数 α 不同时,UCC 策略的缓存性能均高于其他策略.如同文献[26],我们在模拟中采用 $\alpha=0.8$.

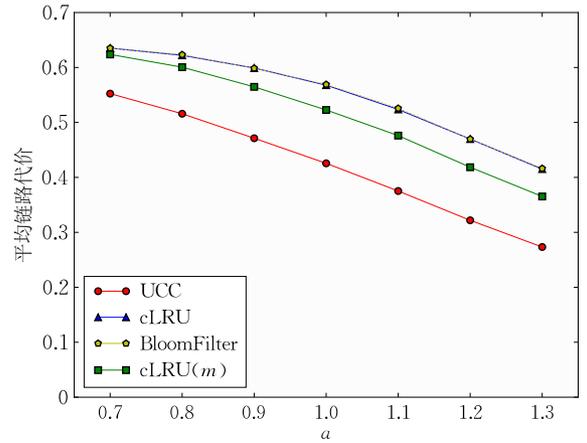


图 12 齐夫分布参数对链路代价的影响

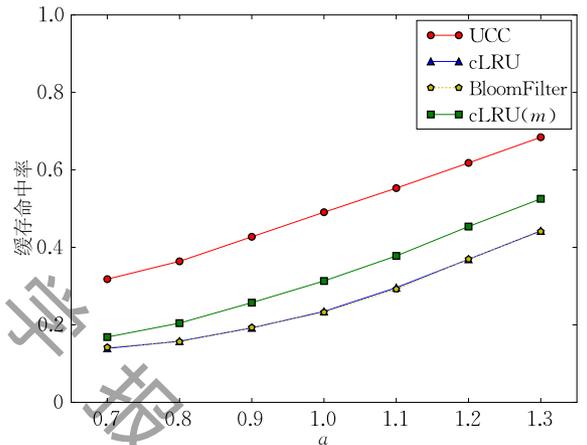


图 13 齐夫分布参数对缓存命中率的影响

5.1.6 缓存容量大小的影响

对缓存系统而言,缓存空间容量是影响缓存性能的关键因素.通常来说,缓存空间越大,缓存命中率越高,缓存系统的效果越明显.之所以使用协作缓存,就是因为单节点的缓存空间有限,通过共享缓存提高整体性能.我们以 UCC 和 cLRU 策略为例对缓存容量的影响进行说明.图 14 和图 15 分别作出了 eNodeB 节点和 UE 节点缓存空间变化对平均链路代价和协作缓存命中率的影响的二维折线图和三维曲面图.折线图中,横坐标表示 UE 节点缓存容量的变化,不同曲线表示 eNodeB 节点缓存容量的变化.横向来看,当 eNodeB 缓存容量不变,UE 缓存容量提升时,平均链路代价呈下降趋势,缓存命中率提升;纵向来看,当 UE 缓存容量不变,eNodeB 缓存容量提升时,平均链路代价和缓存命中率呈同样变化

趋势. 随着缓存容量提高, 缓存的性能就会越明显. 缓存的存储开销和缓存性能是对立的, 如何使用有限的缓存空间发挥更大的缓存性能, 是缓存系统需要解决的问题. 使用协作缓存策略, 目的是将有限的缓存空间共享, 以实现较高的整体性能. 由图可以看出, 在 UE 和 RAN 缓存大小不同的情况下, UCC 策

略的缓存性能均高于 cLRU 策略. 通过三维曲面图也可直接看出相应变化趋势. 总体来看, 与 eNodeB 相比, UE 缓存容量的变化对平均链路代价和缓存命中率的影响较为明显. 因为 UE 缓存容量的增加会直接导致用户自身缓存更多资源, 且 UE 数量众多, 所以 UE 缓存容量的影响更加显著.

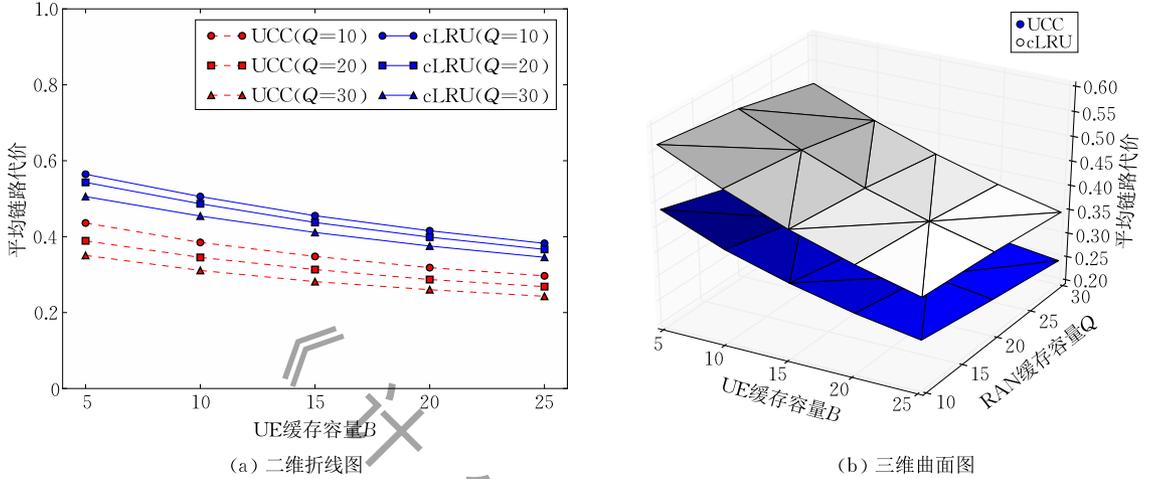


图 14 缓存容量对链路代价的影响

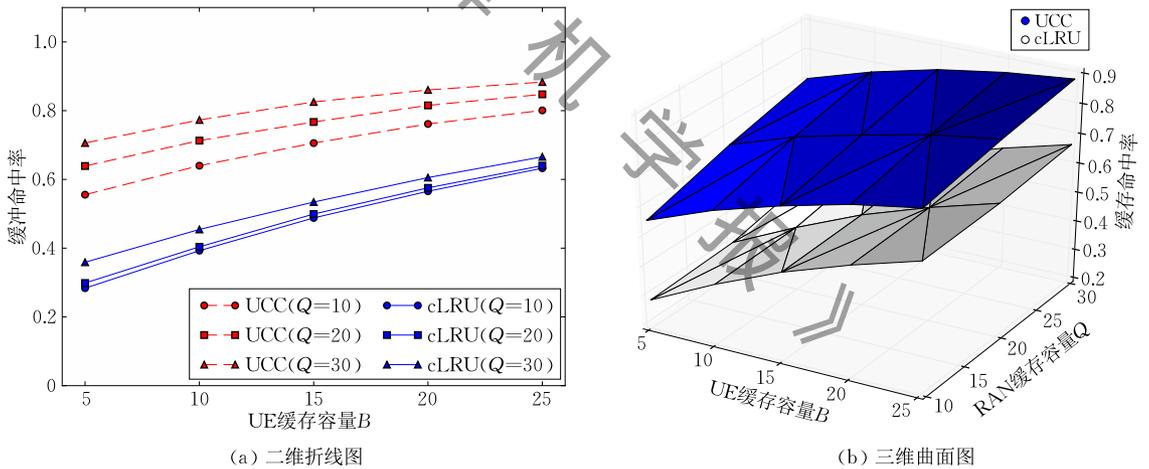


图 15 缓存容量对缓存命中率的影响

5.2 真实数据实验验证

为了验证所提策略在真实移动网络中的性能, 我们根据现网中用户请求的真实数据日志进行了分析验证.

我们使用了中国移动某省公司在 2015 年 4 月 10 日对某真实部署的 LTE 系统中用户 HTTP 请求的测量数据. 测量记录包括时间戳、请求资源 URL、资源大小等字段, 请求总数为 1 316 382, 用户总数为 2670. 该系统包含 EPC 和 3 个 LTE 基站, 但记录信息中没有用户所属基站信息, 我们假定用户均匀接入该基站, 以便进行验证. 我们分别选

取了轻负载时间段(01:00~01:30)和重负载时间段(10:10~10:40)进行分析. 轻负载时间段有 109 个活动用户请求总数为 3984; 重负载时间段有 523 个用户, 请求总数为 37 384. 图 16 作出了两个时间段资源请求排名与请求数的对应关系, 可以看出, 资源请求率基本符合 Zipf 分布. 轻负载时间段和重负载时间段实际资源请求数不同, 轻负载时资源请求数明显低于高负载, 但它们的分布是近似一致的.

由于测量数据中 HTTP 请求大小通常较小且用户间请求分布不均匀, 我们根据真实请求资源的

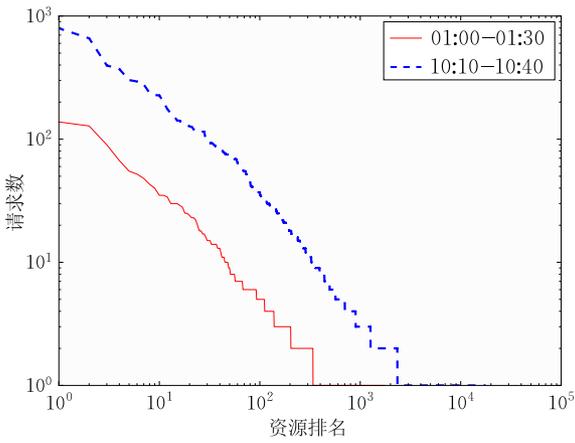


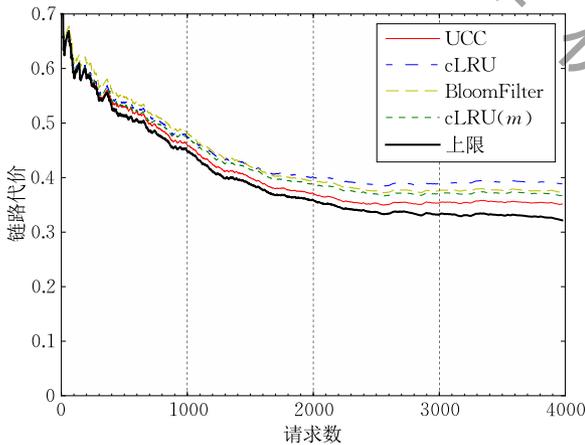
图 16 请求率统计

总体积的百分之一体量级, 设定 UE、eNodeB、EPC 分别具有 10KB、100KB、10MB 的缓存空间, 以达到缓存容量瓶颈, 从而对缓存替换算法性能进行对比. 起始状态缓存为空. 我们根据真实的用户请求记录进行模拟, 通过用户获取资源的来源分析, 从平均链路代价和缓存命中率两个方面对比 UCC 和其他缓存策略, 并作出了缓存“上限”进行对比. 缓存“上限”

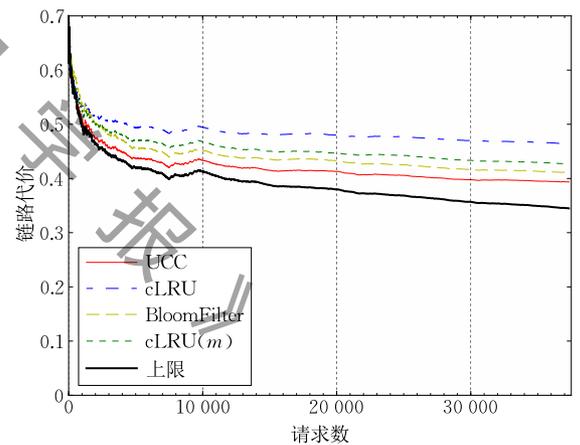
表示当重复请求的资源一定缓存命中时能达到的缓存效果. 它反映了资源请求的重复情况, 是缓存命中率理论上能达到的最高值, 链路代价的理论最小值.

5.2.1 平均链路代价

图 17 分别作出了轻负载和重负载时间段平均链路代价的变化情况. 轻负载时间段, UCC 策略最终链路代价为 0.35, cLRU 策略为 0.39, BloomFilter 策略为 0.37, cLRU(m) 策略为 0.37, 上限为 0.32; 重负载时间段, UCC 最终链路代价为 0.39, cLRU 策略为 0.46, BloomFilter 策略为 0.41, cLRU(m) 策略为 0.43, 上限为 0.35. 初始阶段, 由于缓存为空, 所有请求都从外部获得, 链路代价最大; 随着缓存内容的增多, 平均链路代价降低. 轻负载时间段, 由于缓存容量相对充裕, 各缓存策略性能差距不明显, 但 UCC 策略优于其他策略. 重负载阶段, 由于缓存空间有限, 不同策略性能差距拉大, UCC 策略仍优于其他策略. 总体而言, 轻负载时的平均链路代价优于重负载.



(a) 轻负载时间段



(b) 重负载时间段

图 17 平均链路代价

5.2.2 缓存命中率

图 18 作出了轻负载时间段和重负载时间段缓存命中率的变化情况. 由于初始缓存为空, 缓存命中率从 0 开始变化. 轻负载时间段, UCC 最终缓存命中率为 53.7%, cLRU 策略为 50.9%, BloomFilter 策略为 49.5%, cLRU(m) 策略为 52.2%, 资源重复率为 56.0%; 重负载阶段, UCC 最终缓存命中率为 49.9%, cLRU 为 44.6%, BloomFilter 策略为 46.3%,

cLRU(m) 策略为 46.2%, 资源重复率为 53.4%. 与平均链路代价类似, 轻负载时的缓存命中率总体高于重负载. 值得注意的是, 重负载阶段 BloomFilter 策略的相对缓存性能有所提高, 这是因为重负载阶段单次请求的资源数增加, 而 BloomFilter 过滤掉了该部分资源, 因此缓存性能较为显著. 轻负载阶段, UCC 策略的缓存性能均明显优于其他策略.

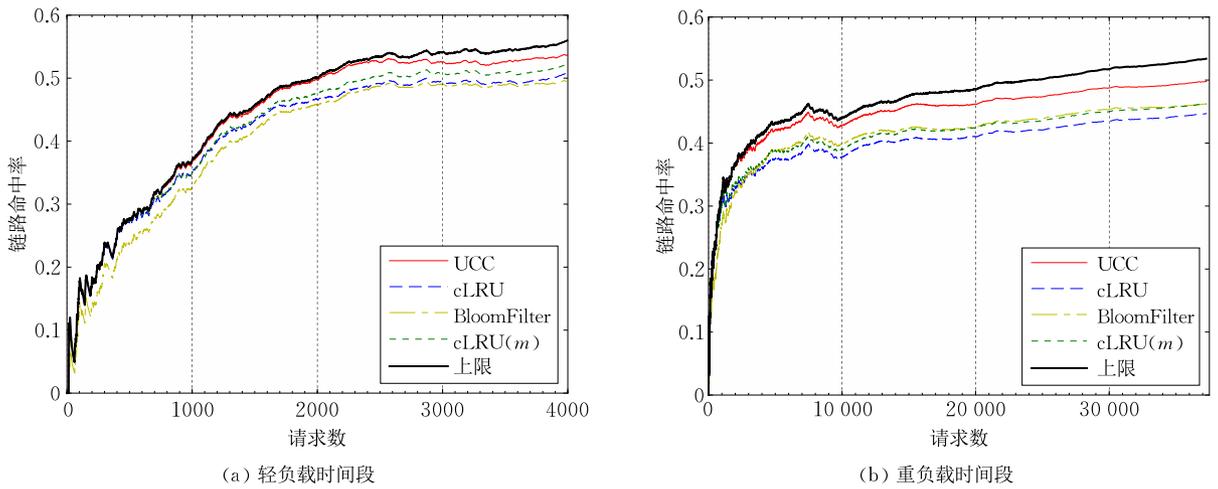


图 18 缓存命中率

5.2.3 协作交互的代价

协作交互的代价可以用交互的次数来粗粒度衡量。协作交互信息主要由如下三部分构成：(1) UE 最初请求某一资源的请求报文和服务器的重定向回复；(2) UE 向资源所在的源节点发送请求报文；(3) 通知中间节点 eNodeB 或 EPC 进行被动缓存。其中，部分(1)理论上必需的。根据文献[31]，资源的标识信息(包括哈希值、大小等)的大小在 40 字节以内。不失一般性，为了更定量地比较不同缓存策略的协作交互代价，我们忽略交互报文的头部、分片等负载，近似地取一次交互信息大小为 40 字节，计算协作交互信息占总传输数据量的比重作为协作交互代价。图 19 作出了协作交互代价的对比情况。

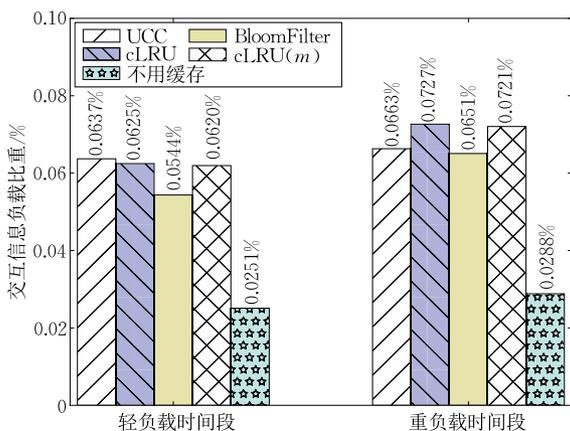


图 19 协作交互代价对比

总体来看，协作交互负载相对不重。cLRU 策略和 cLRU(m)策略的交互代价基本一致，而 BloomFilter 策略的负载最低，这与 BloomFilter 策略通过过滤单次请求的资源来减小磁盘写入次数的设计思想相符。轻负载时，UCC 的交互控制负载略高于

cLRU；而高负载时间段 UCC 的交互控制信息负载低于 cLRU。对于 cLRU 策略，当 UE 从 Internet 获取资源时，EPC 一定会缓存，在计算协作交互代价时，不计入对 EPC 进行缓存控制的交互信息。低负载时，由于初始缓存空间为空，eNodeB 在空间未满足之前，缓存全部资源文件，相比于 cLRU 策略，UCC 计算了对 EPC 的控制信息，所以总体交互信息负载略高于 cLRU。重负载时，缓存替换的次数增多，对于 cLRU 策略，如果协作缓存空间不存在某资源，一定会缓存该资源，而 UCC 则根据 *Utility* 有选择地进行缓存，交互信息负载低于 cLRU 策略，甚至接近 BloomFilter 策略。

5.3 实验评估小结

本节通过模拟仿真和基于真实数据的验证实验，分析了 UCC 策略在移动 CDN 分层协作缓存系统中的效果。由于真实系统中用户请求不均匀，没有充分参与协作缓存；而模拟仿真数据严格按照齐夫分布产生，每个用户均进行了充分的资源请求，所以模拟仿真中 UCC 策略的效果更加显著。在真实系统不同负载情况下，相比其他缓存策略，UCC 策略仍然具有明显优势。总体而言，UCC 策略能实现更低的平均链路代价和更高的缓存命中率，适用于移动 CDN 分层协作缓存系统。

6 结论与展望

本文针对当前移动 CDN 分层协作系统中的缓存放置问题，以最大化节约的链路代价为目标建立分层协作缓存模型并进行分析，基于贪心策略提出了分层协作缓存策略 UCC，并通过模拟仿真和真实

数据实验进行验证. 结果表明, 所提策略适用于移动 CDN 分层协作缓存系统, 相比其他目前广泛使用的缓存策略, 所提策略具有明显优势.

移动 CDN 分层协作缓存系统仍存在值得探讨的问题: (1) 用户的请求排名如何获取. 当前已经存在基于用户的资源访问历史, 使用大数据方法获取用户兴趣, 预测用户请求的技术; 也存在在线实时获取用户兴趣的方法^[34]. 本文假设用户兴趣是已知的, 基于日志数据进行验证. 然而, 该类获取用户兴趣的策略是否适用于移动 CDN 分层缓存系统仍待进一步研究; (2) 本课题模型中的链路代价, 是综合了实验和带宽的量纲. 为简化模型, 将链路代价设为已知. 然而, 在实际的移动核心网络中, 采用何种方式计算链路代价, 在移动 CDN 的实际部署过程中仍需解决; (3) 本文所述模型是相对静态的, 旨在解决移动 CDN 系统中内容协同调度和缓存替换问题. 而用户的移动将引入一系列复杂问题, 例如用户在基站间切换. 利用移动网络的移动管理单元^[44], 文献^[14]提出一种切换时保持传输的机制. 除采用周期性地缓存补偿更新方法外, 如何处理用户移动时引起的无线网络状态的变化, 如何确保用户移动过程中的服务体验和缓存效益的最大化都值得研究. 此外, 移动 CDN 也增加了用户流量窃听^[45]等安全风险, 这些问题需要我们在后续工作中进行探讨.

参 考 文 献

- [1] Li T, Wang K, Xu K, et al. On efficient offloading control in cloud radio access network with mobile edge computing// Proceedings of the IEEE International Conference on Distributed Computing Systems. Atlanta, USA, 2017; 2258-2263
- [2] Cai Ning. Research on the location of the CDN edge node deployment. Telecom Engineering Technics and Standardization, 2015, 10: 71-76(in Chinese)
(蔡宁. CDN 边缘节点部署位置下沉的研究. 电信工程技术与标准化, 2015, 10: 71-76)
- [3] Chen Xue-Liang, Chen Ge, Zhuang Yi-Rong, et al. Study on the key technology of CDN in LTE mobile network. China Internet, 2015, 5: 22-27(in Chinese)
(陈学亮, 陈戈, 庄一嵘等. LTE 移动网络 CDN 关键技术研究. 互联网天地, 2015, 5: 22-27)
- [4] Wang X, Chen M, Taleb T, et al. Cache in the air: Exploiting content caching and delivery techniques for 5G systems. IEEE Communications Magazine, 2014, 52(2): 131-139
- [5] Bastug E, Bennis M, Debbah M. Living on the edge: The role of proactive caching in 5G wireless networks. IEEE Communications Magazine, 2014, 52(52): 82-89
- [6] Zhou Xue-Feng, Xu Ke, Zhang Lan-Shan, et al. Propagation measurement and cluster analysis of time series in social networks. Journal of Chinese Computer Systems, 2015, 36(7): 1545-1552(in Chinese)
(周雪峰, 徐格, 张蓝珊等. 社交网络的传播测量与时间序列聚类分析. 小型微型计算机系统, 2015, 36(7): 1545-1552)
- [7] Peng Ying, Wang Nao, Wang Gao-Cai, et al. Energy-aware cooperation content distribution strategy based on coalitional game in mobile environment. Journal of Chinese Computer Systems, 2016, 37(7): 1383-1388(in Chinese)
(彭颖, 王淦, 王高才等. 移动环境中基于联盟博弈的能量感知协同内容分发策略. 小型微型计算机系统, 2016, 37(7): 1383-1388)
- [8] Wang Quan, Peng Ying, Wang Nao, et al. Study on energy-aware cooperative content distributed strategy in mobile environment. Journal of Chinese Computer Systems, 2015, 36(12): 2808-2812(in Chinese)
(王权, 彭颖, 王淦等. 移动环境下能量感知的协同内容分发策略研究. 小型微型计算机系统, 2015, 36(12): 2808-2812)
- [9] Psaras I, Sourlas V, Shtefan D, et al. On the feasibility of a user-operated mobile content distribution network//Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks. Macao, China, 2017: 1-9
- [10] Zhong Y, Xu K, Li X Y, et al. ESTRA: Incentivizing storage trading for edge caching in mobile content delivery// Proceedings of the IEEE Global Communications Conference. San Diego, USA, 2015: 1-6
- [11] Zheng Kan, et al. Big data-driven optimization for mobile networks toward 5G. IEEE Network, 2016, 30(1): 44-51
- [12] Taleb T, Ksentini A, Jantti R. "Anything as a Service" for 5G mobile systems. IEEE Network, 2016, 30(6): 84-91
- [13] Silva J, Dias A, Nogueira J. Content-aware prefetching in Over-The-Top wireless networks//Proceedings of the IEEE Symposium on Computers and Communications. Heraklion, Greece, 2017: 515-522
- [14] Xie J, Xie R, Huang T, et al. ICICD: An efficient content distribution architecture in mobile cellular network. IEEE Access, 2017, 5: 3205-3215
- [15] Doppler K, Rinne M, Wijting C, et al. Device-to-device communication as an underlay to LTE-advanced networks. Modern Science & Technology of Telecommunications, 2010, 47(12): 42-49
- [16] Tehrani M N, Uysal M, et al. Device-to-device communication in 5G cellular networks: Challenges, solutions, and future directions. IEEE Communications Magazine, 2014, 52(5): 86-92
- [17] Qiao J, Shen X S, Mark J W, et al. Enabling device-to-device communications in millimeter-wave 5G cellular networks. IEEE Communications Magazine, 2015, 53(1): 209-215

- [18] Liu Shu, Striegel A D. Exploring the potential in practice for opportunistic networks amongst smart mobile devices//Proceedings of the 19th Annual International Conference on Mobile Computing & Networking. Miami, USA, 2013: 315-326
- [19] Dahlin M D, Wang R Y, Anderson T E, et al. Cooperative caching: Using remote client memory to improve file system performance//Proceedings of the 1st USENIX Conference on Operating Systems Design & Implementation. Monterey, USA, 1994: 19
- [20] Hara T. Effective replica allocation in ad hoc networks for improving data accessibility//Proceedings of the IEEE International Conference on Computer Communications. Anchorage, USA, 2001: 1568-1576
- [21] Hara T. Cooperative caching by mobile clients in push-based information systems//Proceedings of the 11th International Conference on Information and Knowledge Management. McLean, USA, 2002: 186-193
- [22] Chow C Y, Leong H V, Chan A. Peer-to-peer cooperative caching in mobile environments//Proceedings of the 24th International Conference on Distributed Computing Systems Workshops. Tokyo, Japan, 2004: 528-533
- [23] Chow C Y, Leong H V, Chan A. Cache signatures for peer-to-peer cooperative caching in mobile environments//Proceedings of the 18th International Conference on Advanced Information Networking and Applications. Fukuoka, Japan, 2004: 528-533
- [24] Chow C Y, Leong H V, Chan A T S. Distributed group-based cooperative caching in a mobile broadcast environment//Proceedings of the 6th International Conference on Mobile Data Management. New York, USA, 2005: 97-106
- [25] Chow C Y, Leong H V, Chan A T S. GroCoca: Group-based peer-to-peer cooperative caching in mobile environment. IEEE Journal on Selected Areas in Communications, 2007, 25(1): 179-191
- [26] Borst S, Gupta V, Walid A. Distributed caching algorithms for content distribution networks//Proceedings of the IEEE International Conference on Computer Communications. San Diego, USA, 2010: 1-9
- [27] Zhang Y, Zhou X, Liu Y, et al. A novel cooperative caching algorithm for massive P2P caches. Peer-to-Peer Networking and Applications, 2013, 6(4): 425-433
- [28] Golrezaei N, Shanmugam K, Dimakis A G, et al. FemtoCaching: Wireless video content delivery through distributed caching helpers//Proceedings of the IEEE International Conference on Computer Communications. Orlando, USA, 2012: 1107-1115
- [29] Liu J, Yang Q, Simon G. Optimal and practical algorithms for implementing wireless CDN based on base stations//Proceedings of the IEEE Vehicular Technology Conference. Nanjing, China, 2016: 1-5
- [30] Gast N, Van Houdt B. Transient and steady-state regime of a family of list-based cache replacement algorithms. ACM SIGMETRICS Performance Evaluation Review, 2015, 43(1): 123-136
- [31] Berger D S, Sitaraman R K, Harchol-Balter M. AdaptSize: Orchestrating the hot object memory cache in a content delivery network//Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation. Boston, USA, 2017: 483-498
- [32] Maggs B M, Sitaraman R K. Algorithmic nuggets in content delivery. ACM SIGCOMM Computer Communication Review, 2015, 45(3): 52-66
- [33] Hefeeda M, Noorizadeh B. On the benefits of cooperative proxy caching for peer-to-peer traffic. IEEE Transactions on Parallel & Distributed Systems, 2010, 21(7): 998-1010
- [34] Li S, Xu J, Schaar M V D, et al. Popularity-driven content caching//Proceedings of the IEEE International Conference on Computer Communications. San Francisco, USA, 2016: 1-9
- [35] Wessels D. RFC 2187: Application of internet cache protocol (ICP), Version 2. IETF, 1997
- [36] Fan L, Cao P, Almeida J, et al. Summary cache: A scalable wide-area web cache sharing protocol. Transactions on Networking, 2000, 8(3): 281-293
- [37] Xu Ke, Zhu Min, Lin Chuang. Internet architecture evaluation models, mechanism and methods. Chinese Journal of Computers, 2012, 35(10): 1985-2006 (in Chinese)
(徐格, 朱敏, 林闯. 互联网体系结构评估模型、机制及方法研究综述. 计算机学报, 2012, 35(10): 1985-2006)
- [38] Zhu Min, Xu Ke, Lin Song. The evaluation method towards the application adaptability of Internet architecture. Chinese Journal of Computers, 2013, 36(9): 1785-1798 (in Chinese)
(朱敏, 徐格, 林嵩. 面向应用适应能力的互联网体系结构评估方法. 计算机学报, 2013, 36(9): 1785-1798)
- [39] Shi Xue-Lin, Xu Ke. Utility maximization model of virtual machine scheduling in cloud environment. Chinese Journal of Computers, 2013, 36(2): 252-262 (in Chinese)
(师雪霖, 徐格. 云虚拟机资源分配的效用最大化模型. 计算机学报, 2013, 36(2): 252-262)
- [40] Xu K, Shen M, Liu H, et al. Achieving optimal traffic engineering using a generalized routing framework. IEEE Transactions on Parallel & Distributed Systems, 2016, 27(1): 51-65
- [41] Breslau L, Cao P, Fan L, et al. Web caching and Zipf-like distributions: Evidence and implications//Proceedings of the IEEE International Conference on Computer Communications. New York, USA, 1999: 126-134
- [42] Yang Chen-Kai, Huang Liu-Sheng, Deng Hou, et al. Two phase based content replica placement in wireless Mesh network. Journal of Chinese Computer Systems, 2017, 38(2): 200-203 (in Chinese)
(杨晨凯, 黄刘生, 邓厚等. 无线 Mesh 网络中的两阶段内容副本分配方法. 小型微型计算机系统, 2017, 38(2): 200-203)

- [43] Lin T, Li H, Xie H, et al. Performance and implications of RAN caching in LTE mobile networks: A real traffic analysis // Proceedings of the IEEE International Conference on Sensing, Communication, and Networking. London, UK, 2016: 1-9
- [44] Zuniga J C, Bernardos C J, Oliva A D L, et al. Distributed mobility management: A standards landscape. IEEE Com-

munications Magazine, 2013, 51(3): 80-87

- [45] Xu Ke, Zhao Yu-Dong, Chen Wen-Long, et al. Paradingm-based routing & switching system for data interception attacks. Chinese Journal of Computers, 2017, 40(7): 1649-1663(in Chinese)
(徐格, 赵玉东, 陈文龙等. 防御数据窃听攻击的路由交换范式体系. 计算机学报, 2017, 40(7): 1649-1663)



GE Zhi-Cheng, born in 1993, M. S. His research interests include Internet of Things and content delivery network.

XU Ke, born in 1974, Ph. D., professor, Ph. D. supervisor. His research interests include Internet architecture, high performance router, P2P network, Internet of Things and network economics.

CHEN Liang, born in 1972, senior expert in information

and network security. Her research interests include Internet architecture, hacking and defense technologies, Internet of Things and network security of industrial control system, cryptography.

LI Tong, born in 1989, Ph. D. His research interests include mobile cloud computing and transport protocols.

YAO Long, born in 1976, Ph. D. candidate. His research interests include Internet of Things and information centric network.

SHEN Meng, born in 1988, Ph. D., lecturer. His research interests include Internet of Things and privacy protection.

Background

This paper studies the caching strategy in the field of edge computing and CDN. Mobile CDN is an enhanced content delivery network which put the traditional CDN nodes to the LTE core network and radio access network side. It is a special hierarchical cooperative cache system, including EPC cache, eNodeB cooperative cache and UE cooperative cache. Early works describe the architecture of the specific system and demonstrates the potential advantages of deploying cooperative caching in eNodeB and UE, but does not study what content distribution and caching replacement strategy should be used for the three-tier caching system. Other papers study cooperative cache algorithms in different scenarios, including ad-hoc cooperative caching, single-plane cooperative caching, but they did not address the hierarchical collaboration caching issues in mobile CDN scenarios. Besides, the performance of the cooperative version of the traditional caching algorithms in the mobile CDN scenario needs to be verified. This paper establishes an optimization model for mobile CDN hierarchical cooperative cache system, which aims to minimize the overall cost of user access to resources. Based on the greedy strategy, this paper proposes a heuristic hierarchical Utility-based Cooperative Cache (UCC) strategy, which makes the cache decision based on the global utility value of the cache resources to reduce the cost of user access

to resources as much as possible. Simulation results show that the proposed strategy reduces the link cost by 10% compared to the cooperative list-based caching strategy (cLRU(m)), with a 20% increase in cache hit ratio. Furthermore, the verification experiment based on real data is designed. The results show that the proposed strategy still has obvious advantages compared with other widely used strategies under conditions of different load.

This work is mainly supported by the National Science and Technology Major Project (No. 2015ZX03003004), which named as Research and Experiment of Mobile Content Distribution Network Technology. The core value of the project is to extend CDN to the last miles, adapting it to the SDN/NFV architecture of the future network, and effectively collaborating with the IP network to maximize the traffic flow, improve user experience and service load capacity. This paper aims to solve the caching replacement problem, which is one of the key issues of mobile CDN. The results of our team in this research area include CDN edge storage pricing strategy, cross-source P2P video acceleration platform and Internet architecture. This paper is also supported by the National Key R&D Program of China (No. 2018YFB0803405) and the National Natural Science Foundation of China (No. 61472212).