

附件 2:

哈爾濱工業大學(深圳)

毕业设计（文献）开题报告

题 目:基于信息新鲜度的边缘缓存更新机制研究

专 业 通信工程

学 生 邹清林

学 号 180210405

指导教师 罗晶晶

日 期 2021/11/25

哈尔滨工业大学（深圳）教务部制

说 明

一、开题报告主要内容

1. 课题来源及研究的目的和意义

1.1 研究背景

几十年来，网络关键的性能指标一直是端到端延迟、吞吐量。然而，传统的性能度量指标，如吞吐量和时延，并不能有效地刻画系统中的信息新鲜度，故不再适用于面向实时状态更新类应用的网络设计。传统的数据通信更多关注的是信息传输的时延，即数据从发送时刻到被接收时刻之间的时间间隔；而实时更新应用则更关注信息的时效性，即数据从产生时刻到被使用时刻之间的时间间隔。

与人类年龄相似，数据产生后，随着时间流逝，数据信息的陈旧程度也随之单向增大，不断“老化”。为了使用户有良好的感知体验，信息产生后应尽可能及时地被使用，确保使用的信息尽可能新鲜的。尤其对于实时监测系统(如智能驾驶)和状态更新系统(如工业控制)的部署与应用，信息新鲜度至关重要。

正在逐步普及于网络服务的边缘缓存机制就与信息新鲜度紧密相关。对于一个远程网络服务系统，由于从服务器直接提供服务给客户有高延时高成本等弊端，通常需要在服务器和客户之间用缓存来储存可能提供给用户的内容。与传统缓存不同，边缘缓存将传统的缓存方法和机制集成到基站中，将内容存储移到更靠近最终用户的地方。

相较于传统基站，使用边缘缓存的基站更接近用户，使用更方便。然而基站通常容量较小，通常应该储存使用频率更高的资源。因此产生了边缘缓存的更新问题：信息一般随着时间的流逝而新鲜度下降。为了保证用户得到信息的新鲜度，希望更频繁地在边缘缓存中更新信息。但是由于网络资源有限，如果缓存都以较快的速率进行信息更新，将会导致网络负载较大，甚至导致网络拥塞。另外，用户请求的信息多种多样，往往超出了边缘缓存容量限制。因此，在网络资源受限的情况下，需要考虑一种基于信息新鲜度的边缘缓存更新机制。

1.2 研究意义

当某些内容变得流行或不流行时，它们可能会被添加到边缘缓存中或从缓存中删除。因此产生了边缘缓存的更新问题：一方面，内容在没有更新的情况下存储在缓存中的时间越长，缓存内容变得过时的风险就越高。因此，

我们希望经常刷新缓存的内容，但这会导致回程负载更高。另一方面，基站的存储容量是有限的，不能存储服务器中所有的信息。

优化基站的缓存机制可以降低系统的运行成本，并且减小链路负载。面如何对其内容调度进行优化，该问题已经有十数篇文献提出了相关方案。然而，考虑实际应用的缓存优化要同时考虑：信息的新鲜度、信息的流行度、新鲜度和流行度的时变性、基站下载内容的成本、基站的内容更新能力等因素，而之前的研究未考虑全面，实际性能不理想。

2. 国内外在该方向的研究现状及分析

对于边缘缓存更新策略的研究正随着相关的探究而更加深入。

最初的设计方案在考虑策略时只关注了内容的流行度。2014 年 P. Blasco 和 D. Gndz 与 2015 年 Kader 的研究表明，可以通过基于学习的算法来估计内容的流行度 [1][2]。2016 年，Kiskani 和 Sadjadpour 发表了一种更新策略，其中考虑了具有一定流行度的内容 [3]。然而在上述研究中，内容的流行度是不随时间变化的。2018 年，Harath 在其文献中，研究了流行度在随时间变化的条件下缓存的更新策略 [4]，同年，上海交通大学的 N.Zhang 等人提出了一种算法来估计随时间变化的内容流行度 [5]。

随着研究的深入，缓存更新策略开始考虑到内容的新鲜度。来自 IEEE 的两篇文献开始使用信息的新鲜度，而不是内容的流行度 [6][7]。目前的研究方向是将内容的新鲜度和流行度结合起来考虑。在 Kam, Yates, Heydar Abad 三个团队的研究中，他们兼顾了内容的流行度和新鲜度，但是这些研究仍有不足。在 Kam 团队的研究中，忽略了从服务器下载内容的成本 [8]。在 R. D. Yates 的研究中，缓存的更新能力被设置为每个时隙只能更新缓存的一个内容 [9]，而实际上缓存可以更快的速度进行更新。在 M. S. Heydar Abad 的文章中，作者假设缓存容量是无限的，然而实际基站往往受到缓存容量的限制 [10]。可以看出，考虑实际应用的缓存优化要同时考虑很多因素，而之前的研究未考虑全面。2020 年，Uppsala 大学的 Ahani 和 Yuan 提出了一种涵盖相对全面因素的缓存更新算法，有一定指导意义 [11]。

3. 主要研究内容

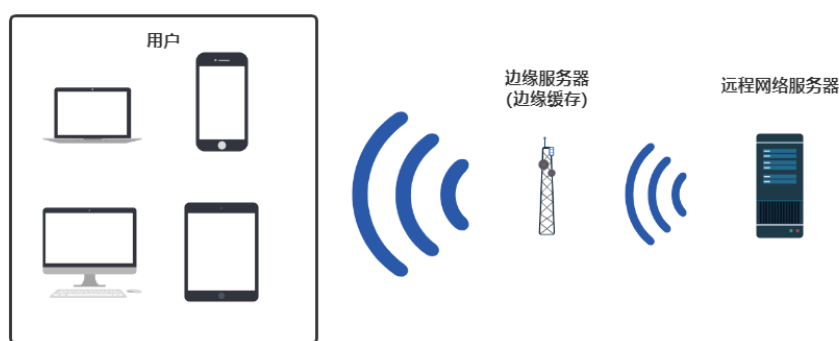


图 1-1 边缘缓存模型

项目主要内容为边缘缓存更新机制的研究，拟设计一种缓存更新机制，在减小回程负载的条件下保证内容的新鲜度。该机制在缓存容量受限的条件下，通过控制边缘缓存的更新来减少回程负载。

考虑到信息的新鲜度以及流行度，以及有限的缓存容量和用户下载的策略，关于边缘缓存更新机制的探究可以转化为一个数学优化问题。

对于得到的更新机制，应该具有在给定缓存条数，更新权重，时间等因素的前提下得到一种更新策略的能力，即下一时刻应该更新多少缓存，从服务器下载多少内容以及哪些缓存不更新。技术指标为在给定条件并保证回程链路负载下，使用设计的边缘缓存更新机制得到的内容的新鲜度。

项目拟采用数学分析软件运行，在对更新问题进行充分调研后，在分析软件中对基站系统进行建模，将缓存更新机制在软件中进行仿真验证并与已知的算法进行性能比较，为进一步深入研究积累经验。相比于已有的研究，本课题综合考虑了内容新鲜度和流行度以及有限的边缘缓存容量，使得结果更具有实际意义。

3.1 新鲜度和信息年龄 AoI

新鲜度指消息的新鲜程度，通常意义上，刚刚产生的信息新鲜程度高，产生很久的信息新鲜程度低。为了量化消息的新鲜度，采用信息年龄的概念。

信息年龄可以用来衡量一个网络中信息的时效性，信息年龄的含义是指从信息产生到当前时刻为止的时间差，和时延最大的区别在于信息年龄不仅仅包含信息的传输时延，还包括信息在源节点的等待时间以及在目的节点的停留时间。对于包含多个源节点的信息更新网络，假设每个源节点需要收集信息，并汇聚到同一个目的节点，则在目的节点包含每个源节点的信息年龄。

在本项目中，信息年龄 AoI 定义为接收的信息从生成到被接收经过的时间，如果这段时间很长，说明信息从生成到被接受经过了很久，信息的新鲜度很低，通常需要更新或者从缓存中删除。

3.2 流行度

流行度，是反映一条信息近期热度的指标。在网络服务模型中，远程服务器有数以万计的信息，而客户可能会请求到任意一条信息。被请求得更频繁地信息拥有更高的流行度，更值得去下载到基站缓存中。然而基站的缓存通常是有限的，因此需要根据实际情况提出相应的更新策略。

4. 研究方案

本项目的研究分为以下阶段：分析优化问题、提出求解方法、仿真验证、撰写论文。

4.1 分析优化问题

作者希望在缓存容量受限的条件下，通过控制内容的更新策略来减少回程负载。假设边缘缓存中存有一定的信息，在下一个时刻，对每一条缓存，基站可能的选择有三种：保持原有信息不更新、用户请求该信息，向用户提供下载内容、该信息在缓存中原不存在但被认为需要写入缓存，基站向远程服务器下载该内容。

在考虑信息的年龄、新鲜度和流行度的基础上，基站的选择还应受到以下约束：信息使用的存储空间小于或等于基站缓存容量；如果内容在缓存中，那么它一定有大于等于 0 的 AoI。如果给基站的三种选择都赋予一项成本，就可以表征相应选择的代价，例如不更新就会使得信息过时，存在消息过时的成本。那么，在本项目中，缓存更新问题可以表述为一个优化问题。将三种从服务器到基站的内容下载 $C_{\text{下载}}$ 、内容更新 $C_{\text{更新}}$ 以及不更新的内容过时成本 C_{AoI} 相加为总成本，内存策略的优化问题即为在约束条件下使得该总成本最小。

$$C_{\text{下载}} + C_{\text{更新}} + \lambda C_{\text{AoI}}$$

λ 为权重系数， $\lambda \in [0,1]$ ， λ 越小表明系统越倾向于不更新，0 的时候表示系统不更新缓存。

得到的算法还应该可以计算更新后的时刻缓存内容的 AoI。信息更新策略还与节点排队策略有关，这里采用的排队策略是先入先出。

4.2 提出求解方法

在数学规划问题中一部分变量或者全部变量为整数变量的话，该数学规划问题就属于整数规划问题。先前的研究表明，边缘缓存的问题可以被公式化为一个整数线性规划 (ILP) 问题，并且被证明是一个 NP 问题，可以在有限时间内求解。

关于如何求解整数规划问题，在数学规划这门课中有涉及。通常情况下，求解整数规划问题有分枝定界法、割平面法、穷举法等。穷举法在变量很多

时可能耗费大量时间，分枝定界法对有约束条件的优化问题的所有可行解空间恰当地进行系统搜索，可用于解整数规划问题，但是对变量个数多的问题不能快速得出结果，割平面法也有相同的缺点。因此需要考虑一种在变量很多的线性规划问题下计算快的方法。

考虑基于单纯型法提出的列生成算法。其思路大概如下：先把原问题 **MP** 简化到一个规模更小（即变量数比原问题少的）的简化问题 **RMP** 下，在 **RMP** 上用单纯型法求最优解，但是此时求得的最优解只是该情况下的，并不是原问题 **MP** 的最优解。此时，需要通过一个子问题去检验在那些未被考虑的变量中是否对原问题有改进的，如果有，那么就把这个变量的相关系数的列加入到 **RMP** 问题的系数矩阵中，经过反复的迭代，直到找不到改进原问题 **MP** 的变量，此时得到原问题 **MP** 的最优解。

4.3 仿真实验验证

基于 **MATLAB** 平台，在先前的分析下提出求解问题并找到求解方法。将求解方法运用编程语言表达出来，并在 **MATLAB** 中尝试运行。

主要包括：对编写好的算法进行性能测试，验证是否可以达到预期标准；对算法进行性能分析，同部分已发表的算法进行比较；探究改进空间，研究改进的可能性。最后，得到相对完整的缓存更新机制。

5. 进度安排，预期达到的目标

2021 年 11 月：调研相关资料，确定方向，完成开题报告。

2021 年 12 月至 1 月：学习有关算法，进行建模，得到优化问题。

2022 年 1 月至 2 月：提出求解方法，编写程序，实现有关算法。

2022 年 3 月至 4 月：进行仿真实验验证，性能分析并加以改进。

2022 年 4 月至 5 月：总结课题研究结果，撰写毕业论文。

6. 课题已具备和所需的条件、经费

所需文献可以在相关网站上查询到，采用计算机仿真方法，由于学校购买了学生查询服务，因此不需要专业的设备以及额外的查询经费。

7. 研究过程中可能遇到的困难和问题，解决的措施

7.1 研究中可能遇到的困难和问题

本课题理论推导以及仿真实验验证，涉及到一部分数学的深入内容，对于查阅资料能力与理解能力有着极大的要求，需要学习查找大量实例、并进行大量的验证仿真。本科阶段对于这种算法方面的仿真实验验证并没有很多的相关实践，编程语言

上来说也比较局限。对于一些相关资料的数学推导,由于它们是算法构成的基础,必须要通过学习来研究清楚。

7.2 解决的措施

之前的课程学习中做过了一些仿真,对于 MATLAB 有一定熟练度,对于代码的读写也有一定基础,所以仿真验证的前期能够更快的完成。加强对代码的学习与练习,学中练、练中学,一定能很好克服代码编写方面的困难。在算法的构建和数学推导上,一方面我会主动去查阅有关资料并且及时总结不懂的部分,另一方面我会及时请教老师和同组学长、同学,学习他们在相关方面上更深入的研究以及新发现的理论,相信在他们的帮助下我可以完成该项目。

8. 主要参考文献

- [1] P. Blasco and D. Gndz, "Learning-based optimization of cache content in a small cell base station," in IEEE ICC, 2014, pp. 1897–1903.
- [2] M. A. Kader, E. Bastug, M. Bennis, E. Zeydan, A. Karatepe, A. S. Er, and M. Debbah, "Leveraging big data analytics for cache-enabled wireless networks," in IEEE GLOBECOM Wkshps, 2015, pp. 1–6.
- [3] M. K. Kiskani and H. R. Sadjadpour, "Capacity of cellular networks with femtocache," in IEEE INFOCOM Wkshps, 2016, pp. 9–14.
- [4] B. N. Bharath, K. G. Nagananda, D. Gndz, and H. V. Poor, "Caching with time-varying popularity profiles: A learning-theoretic perspective," IEEE Trans. Commun., vol. 66, no. 9, pp. 3837–3847, 2018.
- [5] N. Zhang, K. Zheng, and M. Tao, "Using grouped linear prediction and accelerated reinforcement learning for online content caching," in IEEE ICC, 2018, pp. 1–6.
- [6] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal and N. B. Shroff, "Update or Wait: How to Keep Your Data Fresh," in IEEE Transactions on Information Theory, vol. 63, no. 11, pp. 7492–7508, Nov. 2017
- [7] H. Tang, P. Ciblat, J. Wang, M. Wigger, and R. Yates, "Age of information aware cache updating with file-and age-dependent updatedurations," <https://arxiv.org/pdf/1909.05930.pdf>, 2019.
- [8] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, "Information freshness and popularity in mobile caching," in IEEE ISIT, 2017, pp. 136–140.
- [9] R. D. Yates, P. Ciblat, A. Yener, and M. Wigger, "Age-optimal constrained cache updating," in IEEE ISIT, 2017, pp. 141–145.

[10] M. S. Heydar Abad, E. Ozfatura, O. Ercetin, and D. Gndz, "Dynamiccontent updates in heterogeneous wireless networks," in IEEE/IFIP 15th WONS, 2019, pp. 107–110.

[11] G. Ahani and D. Yuan, "Accounting for Information Freshness in Scheduling of Content Caching," ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 2020, pp. 1-6