

面向差异化缓存的隐式协作缓存机制

张佳庚^{1,2}, 郭雷², 王齐¹, 锁志海¹

(1. 西安交通大学网络信息中心, 陕西 西安 710054;

2. 西安交通大学电子与信息学部, 陕西 西安 710054)

摘要: 针对当前信息中心网络 (ICN) 中存在的缓存同质化问题, 提出一种面向差异化缓存的隐式协作缓存机制, 通过对协作组内各节点分配不同种类的缓存任务, 构建了隐式协作组, 实现了相邻节点之间的差异性缓存, 既解决了缓存同质化问题, 又避免了显式协作缓存策略中普遍存在的通信消耗, 从而提升缓存差异性、提高请求的命中率、减少缓存替换次数。

关键词: 信息中心网络; 缓存策略; 隐式协作缓存机制; 差异性缓存

中图分类号: TN914

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2024239

Implicit collaborative caching mechanism for differentiated caching

ZHANG Jiageng^{1,2}, GUO Lei², WANG Qi¹, SUO Zhihai¹

1. Network Information Center, Xi'an Jiaotong University, Xi'an 710054, China

2. Faculty of Electronic and Information Engineering Xi'an Jiaotong University, Xi'an 710054, China

Abstract: Aiming at the problem of cache homogenization in the current information center network, an implicit collaborative caching mechanism for differentiated caching was proposed. By assigning different types of caching tasks to each node in the collaborative group, an implicit collaborative group was constructed to achieve differential caching between adjacent nodes. This not only solved the problem of cache homogenization, but also avoided the communication consumption commonly found in explicit collaborative caching strategies, thereby improving cache diversity, increasing request hit rates, and reducing cache replacement times.

Keywords: information center network, cache strategy, implicit collaborative caching mechanism, differentiated cache

0 引言

在信息中心网络 (ICN, information center network) 中, 内容的请求表现出明显的不均等现象, 即少数内容被大量请求。这种不均等现象在传统 ICN 缓存策略中, 会导致相邻网络节点间易出现严重的缓存同质化问题。尽管在网络全局上, 缓存内容的冗余性有助于降低节点对流行内容的获取代价, 但是, 在局部范围的重复性缓存是对有限缓存

空间的浪费, 会降低节点快速获取其他流行内容的可能性, 对网络性能带来负面影响。

缓存同质化现象造成网络实际缓存的内容数量远低于其可存储容量、降低了节点缓存空间的利用率。特别是, 当网络中新兴热点内容涌现、用户兴趣发生转移时, 同质化缓存会明显阻碍新兴内容的传播, 即当节点请求尚未缓存的内容时, 需要消耗更多的跳数和时间。因此, 解决缓存同质化问题, 对于提升 ICN 传输性能、提高网络服务效能具有重

收稿日期: 2024-10-21

基金项目: 国家自然科学基金资助项目 (No.62172326)

Foundation Item: The National Natural Science Foundation of China (No.62172326)

要意义^[1]。

本文提出了一种面向差异化缓存的隐式协作缓存 (DCIC, differentiated caching-oriented implicit cooperative caching) 机制, 主要贡献如下。

1) 通过对协作组内各节点分配不同种类的缓存任务, 构建了隐式协作组, 实现了相邻节点之间的差异性缓存。

2) 通过在相邻节点之间低传输成本之上构建节点分工协作机制, 以此提升网络命中率。

3) 通过在相邻节点之间建立缓存有效共享的方法, 降低了节点缓存替换次数。

1 相关研究

缓存策略可以根据节点间是否具有协作关系分为协作缓存策略和非协作缓存策略, 其中协作缓存策略又可以根据节点之间是否具有周期性的信息交互分为显式协作和隐式协作。

非协作缓存策略是 ICN 最基本的缓存策略, 如 [1(LCE), 2(RND)] 等。在非协作缓存策略中, 各节点缓存决策过程相同, 相邻节点很容易产生缓存同质化现象。进而导致网络传输性能下降, 主要表现在 2 个方面: 其一是缓存在空间分布上不合理, 相邻的节点容易缓存大量相同的内容; 其二是缓存在时间分布上不合理, 热门内容在其热门时间内被大量缓存, 在热门时间过后又被大量替换, 造成缓存资源的浪费。

近年来, 已有很多研究意识到这一问题, 并提出协作缓存策略来解决缓存同质化问题。其思路是建立节点之间的协作关系, 使节点在缓存时不仅考虑自身的缓存内容, 还要综合考虑其他节点的情况。其中, 显式协作缓存策略的原理是在节点之间建立起固定的通信机制, 以此互相传递信息, 通告彼此缓存的内容, 避免相邻节点的重复性缓存, 达到差异性缓存的目的。其劣势在于通常需要建立额外的定期通信机制, 一方面容易产生较大的网络消耗, 另一方面其通信周期的选择较为困难, 若通信周期过短, 则无效通信过多, 占用大量传输信道资源, 影响正常内容传输; 若通信周期较长, 则节点获取的网络状态等信息会滞后, 引起“扑空现象”, 即节点缓存信息过时失效, 接收到的请求无法响应。

隐式协作缓存策略是指节点之间存在协作关

系, 但并未建立起通信机制的缓存策略。相较于显式协作缓存策略, 隐式协作缓存策略能够避免通信开销。其代表性的策略包括基于网络拓扑、广播机制和状态估计的方法。基于网络拓扑的隐式协作缓存通过挑选拓扑中更占优势的节点来确定缓存位置, 以此避免同质化缓存问题, 但其最大问题在于冲击了 ICN 的扁平化结构, 导致被选中的节点承担了过多的缓存任务, 而其余未被选中的节点则没有被充分利用; 基于广播机制和状态估计的方法根据节点所接收的内容来感知相邻节点的缓存状态, 以避免同质化缓存。但由于其本质是以较小的局部样本来估计网络整体的状态, 其精度难以保证, 对网络性能的提升程度不高^[2]。

本文提出的 DCIC 机制与隐式协作缓存策略不同之处在于, 在避免了频繁通信消耗的前提下, DCIC 机制既保留了 ICN 原有的扁平化结构带来的优势, 又解决了缓存同质化问题, 提升了网络命中率, 降低了节点缓存替换次数, 并且可以用于改进各种非协作缓存策略, 具有很高的泛化性。

2 DCIC 机制概述

2.1 DCIC 原理

传统的非协作缓存策略中, 相邻节点之间未能构建有效的分工协作关系, 节点无法获知相邻节点的存储内容, 在相同的缓存策略下, 相邻节点的缓存内容会出现大量重复, 进而产生了缓存同质化问题。针对这一问题, DCIC 机制通过建立相邻节点对分类内容的分工协作, 对非协作缓存策略进行改进, 实现了不需要额外通信消耗下的差异性缓存。

在非协作缓存策略中, 为每个节点引入内容分类器, 根据内容在全网的统一标识符——内容名对内容进行分类, 将网络中出现的所有内容划分为不同的类别。以每个节点及其一阶邻域 (FON, first order neighborhood) 为单位, 构建隐式协作组, 其中 FON 指节点的周围邻居。DCIC 机制原理如图 1 所示, 通过对协作组内各节点分配不同种类的缓存任务, 使协作组内部能够处理所有类别的内容缓存, 且每个节点处理的任務各不相同, 确保相邻节点之间缓存种类的差异性, 提升了协作组内缓存种类的多样性, 形成一个可覆盖全类别内容、避免缓存同质化、传输时延足够小的协作单元。

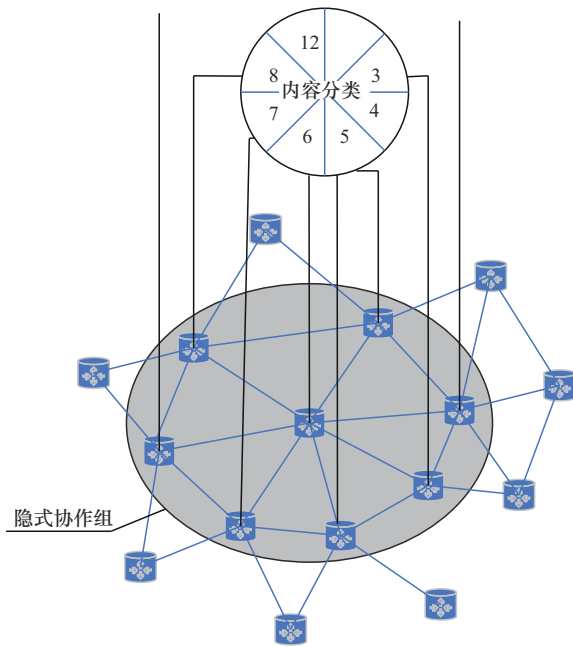


图1 DCIC 机制原理

DCIC 机制对非协作缓存策略的改进主要体现在 3 个方面^[3]。

首先是内容分类过程。在每个网络节点上加入内容分类器，以内容名 (CN, content name) 作为分类算法的输入，以种类标签作为分类算法的输出，使内容分类具有全网统一、完全随机、均等分配的性质，并且能够被所有节点快速一致地识别，为网络的差异性缓存创造基础条件。

其次是缓存种类分配过程。在网络初始化阶段对所有节点，根据其邻居节点缓存状态为其分配缓存种类，约束节点仅能缓存属于此类的内容，并使此节点一阶邻域内缓存种类各异，以提升网络缓存的差异性。

第三是内容识别过程。在缓存放置策略之前加入内容识别环节，对接收到的内容进行种类判别，判断是否与节点的缓存种类一致，确保各节点的缓存内容种类的一致性。

DCIC 机制的本质是建立在相邻节点之间低传输成本之上的节点分工协作和缓存有效共享，以此来实现网络的差异性缓存，并以此提高网络命中率和降低缓存替换次数。

2.2 内容分类

2.2.1 分类算法需求

为了保证内容分类对节点的分工协作提供良好的基础支撑，分类算法需要满足如下 2 个需求。

一是一致性，即所有节点上内容分类器的分类算法要保持一致，相同的内容名应得出相同的类别标签。否则，同样的内容会出现在不同的类别中，导致相邻节点缓存同样的内容，无法保证协作组内完全的差异性缓存，无法实现基于内容分类的协作缓存。

二是均等性，即确保分类后各类别内容在数量的分配上保持平均，避免因各类内容总数不一致而引起的节点之间失衡，部分节点需要缓存的内容过多，造成缓存替换次数过高、缓存内容生存时间过低，而部分节点需要缓存的内容过少，缓存空间没有发挥应有价值。

2.2.2 分类算法设计

为了保证内容分类的一致性，应当采用内容的唯一性标识作为分类的依据，使任意节点对内容进行分类时，能够得到一致性的分类结果。在内容的属性中，考虑到内容的产生/请求时间、流行度等信息都是动态变化的，因此本文以内容名作为分类器的输入参数。

为了保证内容分类的均等性，应当使任意内容被分配至某一种类的概率相等，假设内容类别总数为 C ，则有

$$\forall c \in [1, C], \forall \text{content}, P(\text{content} \rightarrow c) = \frac{1}{C}$$

基于上述考虑，设计如下内容分类器

$$\text{Class}(\text{Content}) = \text{mod} \left(\sum_{i \in \text{CN}} \text{ASCII}(i), C \right)$$

其中， $\text{Class}(\text{Content})$ 为分类器的输出，表示内容所归属的种类标签； $\sum_{i \in \text{CN}} \text{ASCII}(i)$ 为内容名中所有字符的 ASCII 码值之和，实现了内容名从字符串到数值的转换。此分类器以内容名作为分类器的输入，计算内容名中包含的所有字符的 ASCII 码值之和，再对内容类别总数 C 求余，以余数作为分类器的输出。

2.2.3 类别总数 C 的选择

内容类别总数 C 的选择对于缓存机制具有重要的影响。若类别总数过大、远超一阶邻域范围内的节点数量，则节点与其周围邻居无法覆盖所有种类，失去了在其邻域内找到所有内容的可能性，反而降低了网络性能；若类别总数过小，则会因为邻域内节点总数大于类别总数，导致不可避免地出现重复性缓存，降低局部缓存的差异性。因此，理想

情况下,类别总数应当是邻域内节点与其邻居节点的总数,如此即可保证所有类别的内容均有节点进行缓存,且各节点的处理任务各不相同。但实际上,由于网络中各节点的节点度往往并不一致,因此将类别总数取为网络的平均节点度+1,从而尽可能地实现各节点一阶邻域范围的差异性缓存。

2.3 节点缓存种类分配

本文设计了一种基于优先级的轮换分配算法,其基本思想如下^[4]。

为了达到局部上的差异和互补性,在对节点分配缓存种类时,统计其邻居节点中已经具有的种类集合 Set,优先在邻居节点当前缺少的种类中进行选择,使节点邻域内缓存种类各不相同且互补。

若要达到全局冗余和平均分布的目的,则应当尽可能使缓存各个种类的节点数量均等,因此本文提出了内容种类分配的优先级向量 P ,其每个元素 $P(i)$ 表示类别 i 当前被分配的优先程度。初始状态每个种类的优先级相同;每次对节点分配缓存种类时,优先分配优先级最高的种类;在分配之后降低该种类对应的优先级,通过这种方法可以达到各种类轮流分配的目的,使各种类在整体上保持均衡状态。详细的过程如过程 1 所示。

过程 1 节点缓存种类分配过程

STEP1 初始化优先级向量 P

令 $P(i)=1$, 其中 $i \in [1,C]$

STEP2 遍历节点,分配内容种类

for node in network:

if 邻居已有缓存种类集合 Set 为全集 universal,
then 令 node 缓存种类为 j , $P(j)=MAX(P)$, 并令 $P(j)=P(j)-1$

else 取候选种类集合 candidate 为全集 universal 与已有种类集合 Set 的差集; 令 node 缓存种类为 j , $P(j)=MAX(P)$, 并令 $P(j)=P(j)-1$

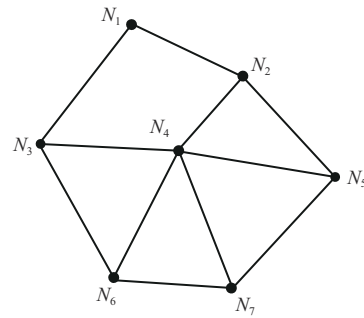
if $P(j)=0$, then $P=P+1$ end if

end if

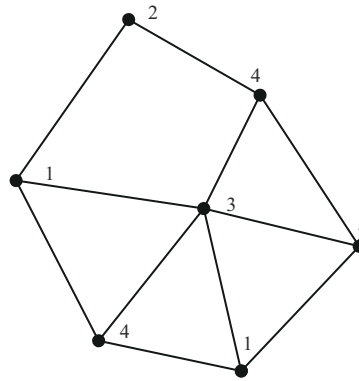
end for

图 2 为种类分配算法示意。图 2(a) 展示了一个简意的示意网络拓扑结构,共包括 7 个节点,其平均节点度为 3.1,取类别总数为 4,初始化优先级向量 $P=[1,1,1,1]$ 。按照上述流程分配节点的缓存种类结果如图 2(b) 所示,节点右上角所标注的数字为节点缓存内容类别序号,从分配结果中可以看出,所

有节点与其邻居节点缓存类别均不相同,实现了网络的差异化缓存。详细的分配过程详如过程 2 所示。



(a) 网络拓扑结构



(b) 缓存种类分配结果

图 2 种类分配算法示意

过程 2 示意网络节点缓存种类分配过程

STEP1 初始化优先级向量 $P = \{1,1,1,1\}$

STEP2 为节点 N_1 分配种类,邻域所含节点 $FON = \{N_1, N_2\}$, 已缓存种类集合 $Set = \{ \}$, 候选种类集合 $candidate = \{1,2,3,4\}$, 所有候选种类优先级相同,随机选择种类 2,更新 $P = \{1,0,1,1\}$

STEP3 节点 N_2 , $FON = \{N_1, N_4, N_5\}$, $Set = \{2\}$, 候选种类集合 $candidate = \{1,3,4\}$, 所有候选种类优先级相同,随机选择种类 4,更新 $P = \{1,0,1,0\}$

STEP4 节点 N_3 , $FON = \{N_1, N_4, N_6\}$, $Set = \{2\}$, 候选种类集合 $candidate = \{1,3,4\}$, 所有候选种类中 $\{1,3\}$ 优先级相同,随机选择种类 1,更新 $P = \{0,0,1,0\}$

STEP5 节点 N_4 , $FON = \{N_2, N_3, N_5, N_6, N_7\}$, $Set = \{1,4\}$, 候选种类集合 $candidate = \{2,3\}$, 所有候选种类中 3 优先级相同,选择种类 3,更新 $P =$

{1,1,1,1}

STEP6 节点 N_5 , $FON = \{N_2, N_4, N_7\}$, $Set = \{3,4\}$, 候选种类集合 $candidate = \{1,2\}$, 所有候选种类优先级相同, 随机选择种类 2, 更新 $P = \{1,0,1,1\}$

STEP7 节点 N_6 , $FON = \{N_3, N_4, N_7\}$, $Set = \{1,3\}$, 候选种类集合 $candidate = \{2,4\}$, 所有候选种类中 4 优先级相同, 选择种类 4, 更新 $P = \{1,0,1,0\}$

STEP8 节点 N_7 , $FON = \{N_5, N_5, N_6\}$, $Set = \{2,3,4\}$, 候选种类集合 $candidate = \{1\}$, 选择种类 1, 更新 $P = \{0,0,1,0\}$

2.4 缓存内容识别环节

DCIC 机制通过在传统非协作缓存策略中增加缓存内容判定环节来发挥其作用, 在缓存放置阶段, 节点判定是否缓存内容包之前, 判定内容种类与节点缓存种类是否一致, 若一致则进入后续缓存放置/替换等流程, 其他流程与传统过程保持一致。以基础的处处缓存 (LCE) 策略和随机缓存 (RND) 策略为例, 应用 DCIC 机制后缓存策略 DCIC-LCE 和 DCIC-RND 在接收到内容包之后, 增加一个内容种类识别环节, 判定内容种类是否与节点缓存内容种类一致, 若一致则缓存该内容(DCIC-

LCE)/生成随机值判定是否缓存该内容^[5]。

3 性能分析

本文提出的隐式协作缓存机制通过构建隐式协作组来实现差异性缓存, 可以实现对内容的均衡分类、确保相邻节点的差异性缓存。为了验证实际的应用效果, 进行了以下分析, 并设计对应仿真实验进行验证。

3.1 内容分类算法均等性验证

为了验证内容分类算法在分配结果上的均等性, 对某多媒体资源平台在 2019—2021 年期间形成的 30 万余条内容请求记录数据, 按照内容分类算法进行处理, 检验各类别下的分类结果的数量和流行度分布情况^[6]。

内容分类算法的类别总数在理想情况下应当设为网络平均节点度加 1, 在本文仿真所采用的网络拓扑中, 节点度取值区间为[2,5], 因此将类别总数分别设置为{3,4,5,6}, 分类后各类所含内容数量和流行度分布情况分别如图 3 所示。

从图 3 中可以看出, 当将内容分别划分为{3,4,5,6}类时, 每次分类结果中各类的内容总数均非常接近。4 种结果的变异系数 (标准差/均值) 分别为 0.39%、0.41%、0.31%、0.41%, 均处于极低水平,

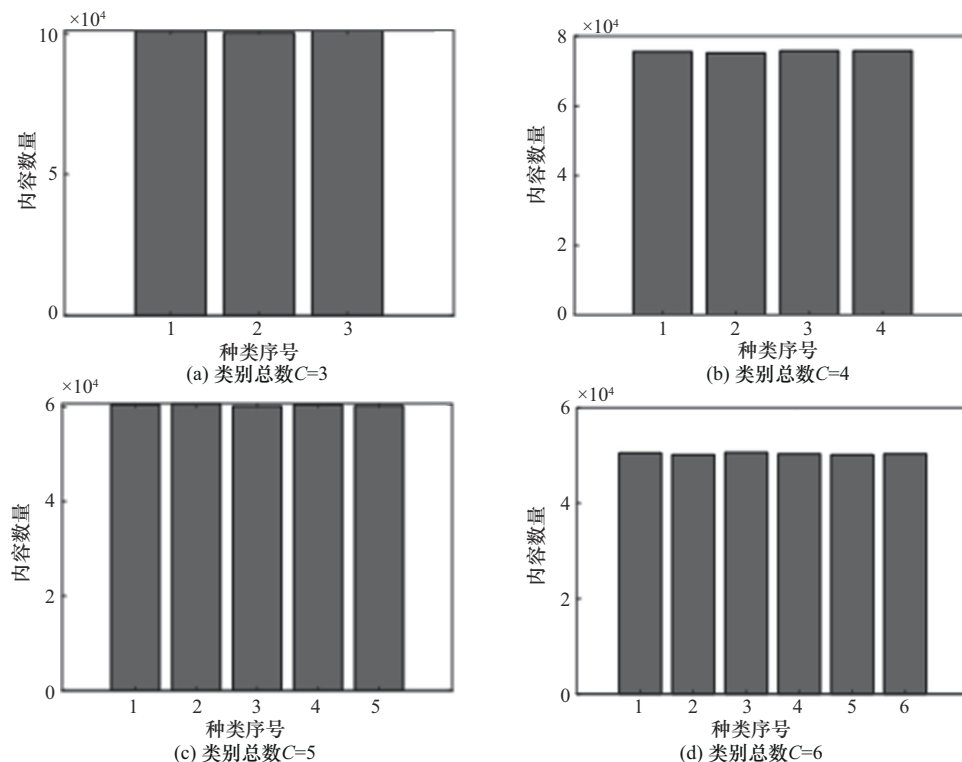


图 3 分类后各类所含内容数量

表明采用内容分类算法能够有效保证各分类结果在数量上的均等性。

表 1~表 4 分别展示了当类别总数分别为 {3,4,5,6} 时各类内容的流行度, 流行度数据为该多媒体资源平台统计的内容访问量。

表 1 各类内容的流行度 (C=3)

内容类别	内容序号	内容流行度
1	0	10^4
2	200	$10^{1.98}$
3	500	$10^{0.15}$

表 2 各类内容的流行度 (C=4)

内容类别	内容序号	内容流行度
1	0	10^4
2	200	$10^{1.99}$
3	400	$10^{0.31}$
4	500	$10^{0.14}$

表 3 各类内容的流行度 (C=5)

内容类别	内容序号	内容流行度
1	0	10^4
2	100	$10^{2.2}$
3	200	$10^{1.91}$
4	300	$10^{1.1}$
5	400	$10^{0.29}$

表 4 各类内容的流行度 (C=6)

内容类别	内容序号	内容流行度
1	0	10^4
2	100	$10^{2.11}$
3	200	$10^{1.89}$
4	300	$10^{0.98}$
5	400	$10^{0.3}$
6	500	$10^{0.05}$

从表 1~表 4 中可以看出, 将所有内容分为 {3,4,5,6} 类时, 各类内容的流行度在很大的范围内取值非常接近, 说明此分类方法能够满足分类算法的均等性需求。

3.2 缓存差异性程度分析

假设节点 $node$ 缓存内容集合为 S_{node} , 所含内

容块的总数为 $|CS_{node}|$ 。为了衡量网络中缓存的差异性程度, 在此引入指标节点缓存差异度 $Diff$

$$Diff_{node} = \frac{|U_{i \in FON(node)} CS_i|}{\sum_{i \in FON(node)} |CS_i|} \times 100\% \quad (1)$$

其中, $FON(node)$ 表示节点 $node$ 所在邻域, $\bigcup_{i \in RON(node)} CS_i$ 为邻域中所有节点缓存内容块的并

集, 其总数 $|\bigcup_{i \in RON(node)} CS_i|$ 表示邻域中缓存的不同内容块总数, $\sum_{i \in RON(node)} |CS_i|$ 为邻域中所有节点缓

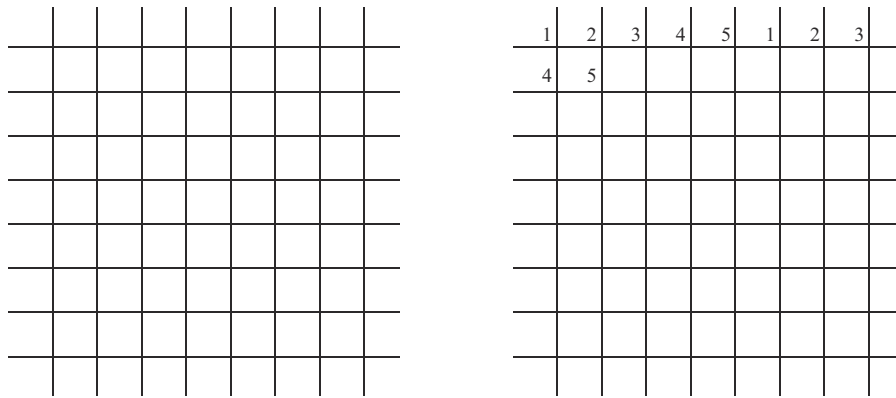
存空间容量总和。 $Diff$ 为节点与其邻居的缓存中不重复内容所占百分比, 其值域为 $[0,100\%]$, $Diff$ 越大, 表示该节点与周围邻居缓存的差异性越高。网络中所有节点差异性指标的均值 $Diff_{avg}$ 可以用于衡量整体网络的差异性缓存程度。

假设在理想网络拓扑情况下, 所有节点的节点度相同, 如图 4(a) 所示, 则根据缓存种类分配算法可得内容类别总数为平均节点度 $4+1=5$, 进行缓存种类分配的结果如图 4(b) 所示。

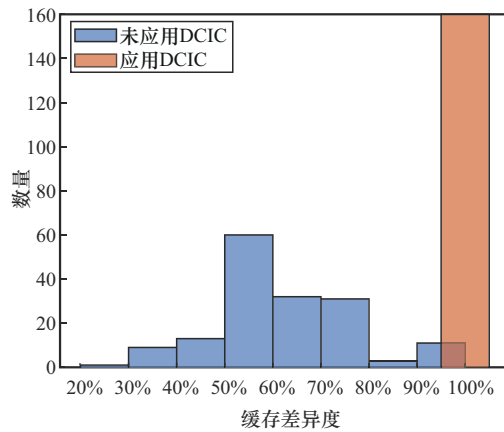
图 4(c) 展示了应用 DCIC 前后节点缓存差异度直方图的对比情况, 从图 4(c) 中可看出, 应用 DCIC 前节点差异度处在较低水平, 大部分节点的缓存差异度均在 80% 以下, 甚至有少量节点的缓存差异度低于 30%, 表明网络中存在较高同质化缓存情况。而应用 DCIC 后, 每一个节点的缓存内容种类与其邻居均不相同, 其 $Diff = 100\%$, 能够有效地保证缓存空间的利用率; 同时, 每一个节点与其所有周围邻居构成的节点集合中覆盖了所有内容种类, 能够保证节点获取不同种类内容的效率; 此外, 所有内容种类网络中的数量完全相等, 保证了网络整体上内容的冗余覆盖。

非理想网络拓扑情况如图 5 所示。图 5(a) 为随机生成的非理想网络拓扑, 图 5(b) 为网络节点缓存种类分类情况, 图 5(c) 为应用 DCIC 前后节点缓存差异度直方图的对比情况。

图 5(a) 中的网络共有 16 个节点, 节点度的取值区间为 $[2,5]$, 平均节点度为 3.1。图 5(b) 是对该网络中各节点分配的缓存内容种类示意, 取内容类别总数为 4, 各节点与其邻居缓存内容种类均不相同。图 5(c) 展示了应用 DCIC 前后节点缓存差异度直方图的对比情况, 未应用 DCIC 时, 节点缓存差异度

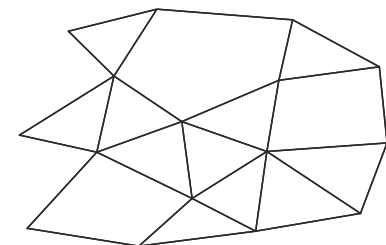


(a) 理想网络拓扑 (b) 缓存种类分配结果

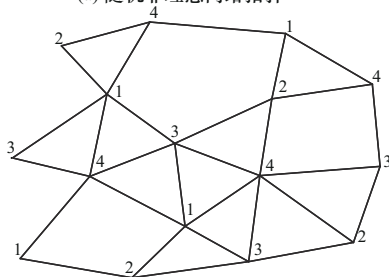


(c) 应用DCIC前后节点缓存差异度直方图对比

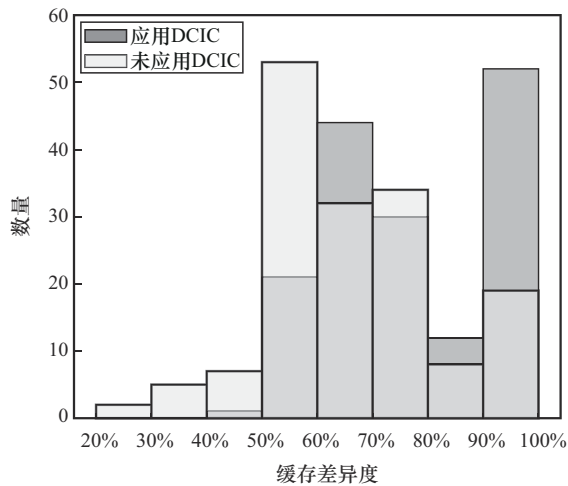
图4 理想网络拓扑情况



(a) 随机非理想网络拓扑



(b) 缓存种类分配结果



(c) 非理想拓扑下应用DCIC前后节点缓存差异度直方图对比

图5 非理想网络拓扑情况

相对较低，大多数节点的Diff低于80%，而网络平均差异度 $Diff_{avg} = 65.68\%$ ，在应用DCIC后，节点缓存差异度有了明显提升，超过一半的节点缓存差

异度高于90%，其均值 $Diff_{avg} = 78.63\%$ ，在前者的基础之上提升了19.72%，可见应用DCIC能够有效提升网络节点的缓存差异度。

4 仿真实验

本文提出的 DCIC 机制是对传统的非协作缓存策略的改进,采用非协作缓存策略中最常见的 LCE 和 RND 这 2 种策略作为性能对比的对象。通过在相同网络环境下分别应用 LCE/RND 与 DCIC-LCE/DCIC-RND 策略,对其命中率、平均时延和缓存替换次数进行对比,以验证 DCIC 机制对于非协作缓存策略能够产生的影响。

4.1 实验参数设置

设定的网络场景为一个扁平化的边缘网络,相邻节点具有互相通信能力,所有节点处在同一个边缘控制器的管控下,节点请求内容时优先在边缘网络内部进行路由寻址,当边缘网络内部无法找到内容时交由控制器向主干网络发送请求,以保证边缘节点能够获取到所需内容。假设网络转发规则为泛洪,所有节点之间传输时延均相等。

边缘网络参数设定如下:节点总数为 100 个,平均节点度为 4,内容总数为 1000,节点缓存容量为 10;内容的请求符合 $\lambda = 1$ 的泊松分布,内容流行度符合 $C = 1.57, \alpha = 2$ 的 Zipf 分布。

4.2 性能参数

4.2.1 平均响应时延

响应时延是节点从请求内容的时刻到其接收到请求内容时刻之间的时间间隔,是网络传输性能的主要体现指标之一。取仿真时间内所有请求响应时延的均值作为网络的平均响应时延 $\text{Delay}_{\text{avg}}$ 。

4.2.2 边缘命中率

边缘网络期望将更多的传输业务在底层边缘节点中进行,实现业务下沉的目的。为了测量边缘网络,定义边缘命中率 (EHR, edge hit ratio) 为

$$\text{EHR} = \frac{n_1}{n_2} \quad (2)$$

其中, n_1 表示请求在边缘节点缓存中得到响应的次数, n_2 表示所有请求的总数。

4.2.3 缓存替换次数

节点缓存空间已满之后,当新的内容需要缓存时,需要将已缓存内容替换为新内容。缓存替换次数 (CRT, cache replacement time) 是网络中所有节点发生缓存替换事件次数的总和。缓存替换对网络带来了大量的计算资源和能源消耗,对于节点来说,大量的缓存替换次数将会对节点的转发业务带来负面的影响。因此希望尽可能减少缓存替换次数。

4.3 仿真结果

对 LCE、RND 和 DCIC-LCE、DCIC-RND 这 4 种缓存策略分别进行 1000 次蒙特卡罗仿真实验,得出网络性能表现如表 5 所示。

从表 5 中可以看出,应用隐式协作扩容机制后与原非协作缓存策略相比 (DCIC-LCE 相对于 LCE、DCIC-RND 相对于 RND),前者与后者在平均时延上基本能够保持一致,但是边缘命中率与缓存替换次数得到了非常大的改善,平均边缘命中率在 LCE 策略对比中从 82.39% 提升至 95.87%,在 RND 策略对比中,平均缓存替换次数则分别降低为原策略的 11.69% 和 10.81%。以上结果说明,应用本文提出的扩容机制能够有效提升网络的缓存能力,增加网络的命中率,极大地减少节点缓存替换次数,显著提升了网络性能。

5 结束语

本文针对当前信息中心网络中存在的同质化缓存与协作缓存通信代价消耗问题,提出一套基于隐式协作方式的缓存扩容机制,通过构建隐式协作组来避免协作通信消耗,保证协作组内部的严格差异性缓存,避免了同质化缓存造成的不良影响,实现对传统非协作缓存策略综合性能的有效提升。

从仿真结果可知,本文提出的机制能够有效地降低缓存替换次数,提升边缘网络命中率,有效地

表 5 仿真实验结果

缓存策略	边缘命中率			平均响应时延			缓存替换次数		
	最大值	平均值	最小值	最大值	平均值	最小值	最大值	平均值	最小值
LCE	77.75%	82.39%	86.25%	10.5	3.40	6.90	25.32	22.19	18.74
DCIC-LCE	93.63%	95.87%	97.88%	10.25	3.50	6.90	6.47	2.59	1.36
RND	81%	84.95%	88.38%	10.42	3.50	6.91	24.28	20.41	17.76
DCIC-RND	93%	95.90%	97.75%	9.40	3.50	6.89	3.99	2.21	1.03

提升了网络的综合性能。

未来的研究将着眼于以下 2 个方面：一是从内容的实际意义出发，更深层次地认知网络状态，在保证平均分类的前提下实现内容分类与内容实际意义的整合，让内容的分类能够发挥更加实际的作用；二是研究在动态网络拓扑中如何实现节点缓存内容种类的有效分配，以保证节点的差异性缓存以及避免缓存种类的冗余性。

参考文献：

- [1] 张鑫. 基于隐式协作缓存机制的 WSN 网络能耗优化算法[J]. 计算机仿真, 2024, 41(2): 425-429.
ZHANG X. WSN network energy optimization algorithm based on implicit cooperative caching mechanism[J]. Computer Simulation, 2024, 41(2): 425-429.
- [2] 王翔. 移动边缘计算中分布式智能服务缓存和资源分配联合优化[J]. 重庆理工大学学报, 2024, 38(15): 219-226.
- [3] 伊鹏, 李根, 张震. 内容中心网络中能耗优化的隐式协作缓存机制[J]. 电子与信息学报, 2018, 40(4): 770-777.
YI P, LI G, ZHANG Z. Energy optimized implicit collaborative caching scheme for content centric networking[J]. Journal of Electronics & Information Technology, 2018, 40(4): 770-777.
- [4] 闫晓瞳, 刘丹谱, 张志龙. 星地融合网络中基于深度强化学习的多业务缓存策略[J]. 无线电通信技术, 2023, 49(5): 875-882.
YAN X T, LIU D P, ZHANG Z L. Multi-type services caching strategy based on deep reinforcement learning in integrated satellite-terrestrial networks[J]. Radio Communications Technology, 2023, 49(5): 875-882.
- [5] 周继鹏, 李祥. 基于深度强化学习的边缘网络内容协作缓存与传输方案研究[J]. 计算机应用研究, 2024, 41(6): 1825-1832.
ZHOU J P, LI X. Deep reinforcement learning based-edge network content cooperative caching and transmission scheme[J]. Application Research of Computers, 2024, 41(6): 1825-1832.

[6] 金宁, 周文倩, 周旭颖, 等. 面向合作博弈及深度学习的节点协作缓存机制[J]. 北京邮电大学学报, 2024, 47(3): 55-61, 68.

JIN N, ZHOU W Q, ZHOU X Y, et al. Collaborative caching mechanism of nodes based on cooperative game and deep learning[J]. Journal of Beijing University of Posts and Telecommunications, 2024, 47(3): 55-61, 68.

[作者简介]



张佳庚 (1990-), 男, 陕西韩城人, 西安交通大学工程师, 主要研究方向为 5G、下一代互联网。



郭雷 (1990-), 男, 陕西西安人, 西安交通大学博士生, 主要研究方向为机器学习、5G。



王齐 (1982-), 女, 陕西杨凌人, 西安交通大学工程师, 主要研究方向为高校信息化、网络舆情。

锁志海 (1971-), 男, 回族, 上海人, 西安交通大学工程师, 主要研究方向为高校信息化、网络舆情。