文章编号：1003－1251(2026)01－0024－07

基于数据冗余包的协同缓存放置策略

西 浓，谭小波，刘菁宇

（沈阳理工大学 信息科学与工程学院，沈阳 110159）

摘 要： 针对命名数据网络(named data networking，NDN)中缓存冗余高、资源利用率低的问题，提出基于数据冗余包的协同缓存放置策略(cooperative cache placement strategy based on data redundancy packet，CCPS-DRP)。该策略先对网络进行社区划分，再利用复杂网络中节点介数确定关键路由节点，将数据冗余包中的相同数据副本存放于其中，并动态管理冗余数据副本，实现关键节点集中管理与动态冗余清理的协同机制，以减少相同数据内容的重复缓存，使路由节点中的缓存容量得到充分利用，降低网络负载。实验结果表明，无论在均匀分布或高度偏斜分布的场景下，还是在严格资源限制或有较高资源的场景下，CCPS-DRP在缓存命中率和平均请求时延方面的性能均明显优于常见的随处缓存策略、概率缓存策略等，其通过协同缓存与冗余控制显著提升了网络性能。

关 键 词： 数据冗余包；协同缓存放置策略；关键路由节点；动态管理；缓存命中率

中图分类号： TP393.07 **文献标志码：** A **DOI：**10.3969/j.issn.1003－1251.2026.01.004

Cooperative Cache Placement Strategy Based on Data Redundancy Packets

XI Nong，TAN Xiaobo，LIU Jingyu

（Shenyang Ligong University，Shenyang 110159，China）

Abstract： A cooperative cache placement strategy based on data redundancy packet(CCPS-DRP) is proposed to tackle the problems of high cache redundancy and low resource utilization in named data networking(NDN). The strategy first performs community division for the network，and then determines key routing nodes by leveraging the node betweenness in complex networks. It stores the identical data replica content from data redundancy packets in these nodes and dynamically manages redundant data replicas，realizing a collaborative mechanism of centralized management for key nodes and dynamic redundancy cleanup. This aims to reduce repeated caching of the same data content，make full use of the cache capacity in routing nodes，and lower network load. Experimental results indicate that CCPS-DRP demonstrates significantly better performance than common caching strategies like the everywhere caching strategy and probabilistic caching strategy in terms of cache hit rate and average request delay，whether in scenarios with uniform distribution or highly skewed distribution，or under strict resource constraints or with abundant resources. By means of collaborative caching and redundancy control，CCPS-DRP notably improves network performance.

Key words： data redundancy packet；cooperative cache placement strategy；key routing node；dynamic management；cache hit ratios

收稿日期： 2025－05－09

基金项目： 辽宁省科技厅“揭榜挂帅”项目(2023JH1/10400093)

作者简介： 西浓(2001—)，女，硕士研究生；谭小波(1977—)，通信作者，男，教授，博士。

命名数据网络(named data networking,NDN)是信息中心网络的核心架构,其以数据为中心,通过内容名称(而非 IP 地址)进行路由,支持网络层内置缓存与消费者驱动的请求 - 响应模式,优势在于高效内容分发、多路径传输及移动性支持。NDN 中缓存信息的存储位置对于网络性能和用户体验具有重要影响,当网络环境和用户需求发生变化时,缓存策略也需进行相应调整以提升其适应性[1]。在 NDN 架构中,缓存策略是支撑其核心优势(如高效内容分发、低时延传输)的关键技术,采用合适的缓存策略可以显著优化网络性能、提高资源利用率[2]。

目前常见的缓存策略包括概率缓存策略(ProbCache)[3]、基于数据请求节点的就近缓存策略(CPCA)[4]、随处缓存策略(LCE)及基于节点重要度和内容类别的缓存放置策略(NICC)[5]等四种。ProbCache 策略根据网络状态动态计算缓存概率,存在计算开销高的问题;CPCA 策略优先在距请求者最近的节点缓存内容,可缩短后续请求的响应路径,但该策略对拓扑变化敏感、需动态更新路径信息;LCE 策略的核心思想是数据包经过的路径上所有节点均无条件缓存副本,存在存储浪费、冗余度高的问题;NICC 策略结合节点重要性与内容类别进行差异化缓存,需多维数据支撑,算法复杂度较高。

近年来很多学者对缓存策略进行了深入研究。Herouala 等[6]提出了分类数据缓存策略,通过深入了解用户行为,将热门请求进行分类,创建了更高效的缓存机制,实现了更高的缓存命中率。Zha 等[7]提出了一种基于动态流行度和替换值的缓存优化策略,当请求内容到达路由节点时,根据当前周期的请求数量和前一周期的流行度计算最新流行度,并根据节点缓存空间占用情况调整缓存阈值。当缓存空间已满时,依据缓存内容的最后请求时间、流行度和传输成本计算替换值,将替换值最低的内容移出缓存空间。与其他传统 NDN 缓存策略相比,该策略有效提高了缓存命中率。Hou 等[8]提出了基于图神经网络(GNN)的缓存策略,首先利用卷积神经网络提取 NDN 节点的时间序列特征,然后采用 GNN 对每个 NDN 节点的内容缓存概率进行预测,最后根据缓存概率排序做出缓存替换决策。文献[9 - 10]亦针对缓存命中率低、网络节点负载重等问题提出了相应的缓存策略与解决方法,一定程度上改善了上述问题。

当前关于数据缓存策略的研究已经取得了一定进展,但仍存在灵活性差、冗余副本过多、资源利用率低、计算复杂度高等问题。为此,本文提出一种基于数据冗余包的协同缓存放置策略(cooperative cache placement strategy based on data redundancy packet,CCPS-DRP),通过动态管理冗余数据副本,减少相同数据内容的重复缓存,使路由节点中的缓存容量得到充分利用,降低网络负载,进而提升网络性能和用户体验。

1　基于数据冗余包的协同缓存放置策略

CCPS-DRP 的核心思想是通过在路由节点之间新增数据冗余包,并为其设定 Tag 字段,以存储冗余数据副本名称。为优化数据存储位置,先对网络进行社区划分,再通过复杂网络中节点介数确定社区内的关键路由节点[11],最后依据就近原则在关键路由节点中缓存冗余度高的数据,从而降低整个网络的缓存冗余度,提升资源利用率[12]。

1.1　数据冗余包的结构与创建

新增数据冗余包的序号与当前路由节点 ID 一致。当数据冗余包转发到其他路由节点时,若路由节点中存在与数据冗余包中相同的数据名称,则将该路由节点 ID 进行存放。冗余数据副本中记录第一个兴趣包到达的路由节点的内容存储库(content store,CS)中数据副本名称,目的是比对其他路由节点是否存在相同的数据副本名称,并存储路由节点 ID。数据冗余包每转发到一个路由节点并匹配到相同数据名称时,冗余次数记录增加 1 次。

创建数据冗余包的步骤如下。

步骤 1:在 tlv. hpp 文件中添加自定义 Tag 字段,定义数据冗余包的序号、路由节点 ID、冗余次数和冗余数据副本。

步骤 2:在 ndn-cxx 文件下新建 hpp 文件,仿照兴趣包结构编写,并引用新增的 Tag 字段。

步骤 3:在 ndn-cxx 文件下新建 cpp 文件,仿照兴趣包结构编写,并实现自定义 Tag 字段的编码和解码。

步骤 4:验证自定义 Tag 字段是否成功加入。

1.2　数据冗余包的生成与转发

在转发平面底层对数据冗余包进行处理。当一个兴趣包到达路由节点时,节点首先查询 CS。

若 CS 中存在匹配名称的数据包,则节点直接返回该数据包,并同步生成数据冗余包以携带当前路由节点的 CS 中所有数据副本名称;若不存在,则根据转发信息库(forwarding information base, FIB)转发兴趣包,且此时仍会触发数据冗余包的生成。此外,当路由节点的 CS 存储不足时,会主动生成数据冗余包,通过 FIB 接口将其转发到其他路由节点进行数据名称比对。如果其他路由节点存在相同数据名称,则认定为冗余数据,将其副本从该路由节点的 CS 中删除,并记录路由节点 FIB 接口。每进入一个路由节点,若其 CS 中存在相同数据名称,则用于记录冗余次数的 Tag 字段值加 1[13]。

当路由节点的 CS 存储不足而生成数据冗余包时,不考虑数据副本的大小,通过统一资源定位符路径的格式查找数据冗余包。因为在路由节点的 FIB 表中已经存有到达指定生产者的数据接口,所以数据冗余包无需存储数据本身,只存放数据名称,而 CS 中存放真实数据。此后,数据冗余包将冗余次数较多的数据名称放置到关键路由节点中,再将冗余包中的该数据名称进行清除,以减少整个网络拓扑结构中冗余的数据副本。

1.3 社区划分

社区划分的目的是使分区后的社区内部连通性更强、社区间的连通性更弱。采用 Girvan-Newman(GN)策略[14]划分社区,即通过迭代优化模块度实现划分,过程如下。

首先清除冗余数据副本,保留数据名称至冗余包,确保网络结构反映真实连接关系。然后将用户抽象为节点,兴趣关系抽象为边(权重由交互强度定义),构建有向图。初始时每个节点为独立社区,逐步合并或调整节点分配,计算模块度增量。模块度 Q 计算式为

$$Q=\frac{1}{2m}\sum_{i,j}\left[A_{i,j}-\frac{k_i k_j}{2m}\right]\delta(c_i,c_j) \qquad (1)$$

式中:m 为总边权重;$A_{i,j}$ 为节点 i 和节点 j 之间边的连接强度;$k_i=\sum_j A_{i,j}$,表示节点 i 的度;c_i 表示节点 i 被分配到的社区;$\delta(c_i,c_j)$ 用于判断节点 i 和 j 是否同属于一个社区,若是,返回 1,否则返回 0。

每次选择使 Q 值提升最大的操作,直至 Q 值无法再优化,最终划分满足社区内高内聚、社区间低耦合的特征。本文实验基于 Karate 网络,其社区划分结果如图 1 所示。

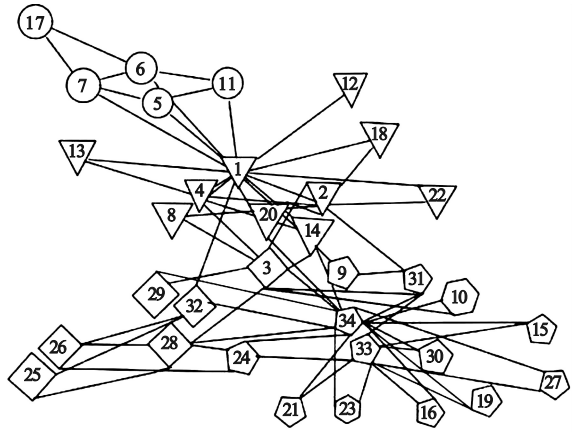

图 1 Karate 网络的社区划分图

Fig. 1 Community division graph of the Karate network

1.4 关键路由节点的确定

在整个网络拓扑结构中,所有冗余数据副本均被清除掉,并将对应的数据名称存放在数据冗余包中,根据度中心性、介数中心性等相关理论确定关键路由节点。

在网络分析中,节点的度中心性是最基础、最直观的中心性度量指标,用于衡量节点在网络中的局部重要性,节点的度数越大,其重要性水平也就越高。

网络内某块中任一节点到其他某块中任一节点的最短路径条数也是用来衡量节点重要性的指标。节点的介数是指所有最短路径中经过该节点的路径数量占比,网络中任意两个节点的所有最短路径中经过某个节点的路径越多,则认为该节点的介数中心性越高。节点 i 的介数定义为

$$B(i)=\sum_{s\neq i\neq t}\frac{n_{s,t}^{i}}{g_{s,t}} \qquad (2)$$

式中:$B(i)$ 表示节点 i 的介数;$g_{s,t}$ 表示从节点 s 到节点 t 的最短路径数量;$n_{s,t}^{i}$ 表示从节点 s 到节点 t 的最短路径中通过节点 i 的路径数量。

本文选择度中心性与介数中心性双高的节点作为关键路由节点,此类节点兼具高连接性和高流量承载能力,是网络性能的核心支撑点。

2 基于数据冗余包的协同缓存放置策略实施流程

NDN 在路由转发过程中,数据的缓存策略由 CS 中设置的缓存替换策略和全局路由缓存放置策略组成。本文将全局路由节点分为关键路由节点和普通路由节点两类,提出的缓存策略网络如图 2 所示。

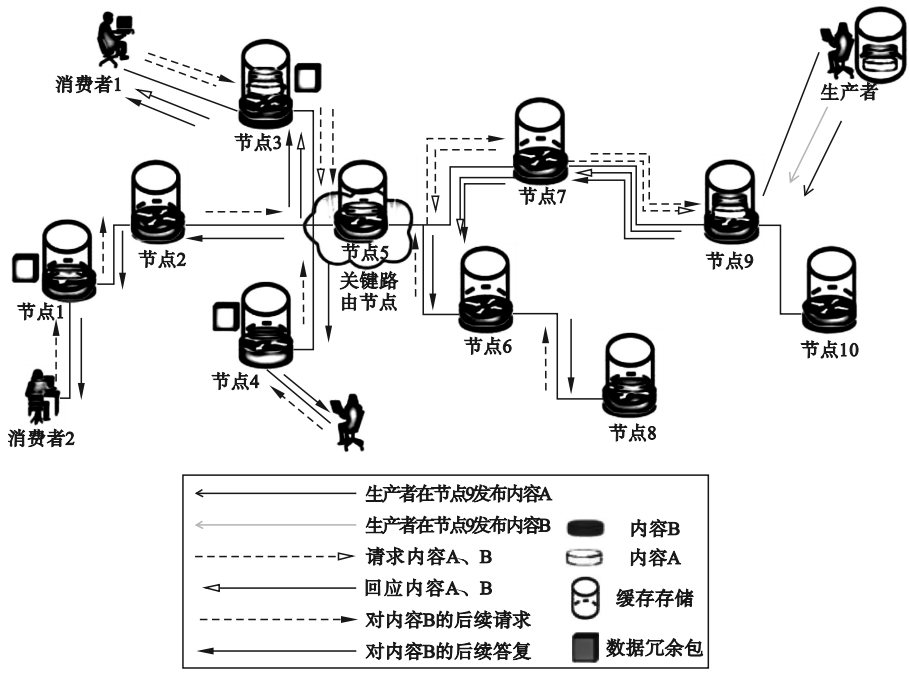

图 2　缓存策略网络图

Fig. 2　Cache policy network diagram

整个缓存过程中,在关键路由节点(节点 5)使用 ndnSIM 仿真软件自带的先进先出替换策略(FIFO)[15]对数据包进行处理。

数据冗余包通过路由节点中 FIB 接口转发到下一个路由节点,依次遍历整个网络拓扑结构中的路由节点[16],以获取冗余次数较高的数据名称,然后数据冗余包会将携带的冗余数据名称就近存储到关键路由节点(节点 5)中[17],之后清除其余普通路由节点中具有相同名称的冗余数据内容[18],从而达到与普通路由节点协同缓存的效果,降低整个网络拓扑结构的缓存冗余度,提升其缓存利用率。数据冗余包在关键路由节点(节点 5)中保留冗余数据时,如发现该路由节点的 CS 存储容量不足,优先保存冗余次数较多的数据内容。数据冗余包的执行与数据包的执行过程互不影响。

消费者发送兴趣包请求数据内容,当兴趣包到达路由节点时,该路由节点会产生一个数据冗余包并对应获取该路由节点的 ID 信息、数据冗余包的序号和该路由节点中 CS 存储的数据名称。

3　实验环境与结果分析

3.1　实验环境及参数设置

实验在 NS-3 网络模拟器和 ndnSIM2.7 仿真平台上进行。采用 Ubuntu18.04 操作系统,网络拓扑基于 Karate 数据集,模拟 Zachary 空手道俱乐部的社交网络[19],包含 34 个节点和 78 条边(见图 1)。实验参数如表 1 所示。

表 1　实验参数表

Table 1　Table of experimental parameters

参数	数值
生产者数量	2
消费者数量	5
兴趣包请求频率(服从泊松分布)/(req·s^{-1})	20
CS 容量大小 C	60 ~ 480
内容单元的数量 N	6 000
缓存容量比 $R(R=C/N)$	0.01 ~ 0.08
用户对内容请求的概率 a(服从 Zipf 分布,即 Zipf 分布系数)	0.6 ~ 1.8(默认为 0.8)
仿真时间/s	120

实验选取 a 的范围为[0.6,1.8],覆盖了从均匀分布($a=0.6$)到高度偏斜分布($a=1.8$)的场景,用以验证缓存放置策略在不同内容热度分布下的适应性。实验设定 R 的范围为[0.01,0.08],模拟实际 NDN 中边缘节点资源受限的场景,$R=0.01$ 时对应严格资源限制,$R=0.08$ 时对应较高资源场景,用以验证缓存放置策略在资源紧张与宽松环境下的鲁棒性。

3.2　性能评价指标

本文采用缓存命中率和平均请求时延作为缓存放置策略性能的评价指标。缓存命中率是指在用户的请求到达 NDN 节点时,节点能够从其本地

缓存中直接提供请求内容(数据块)的请求数量与总请求数量的比值,用于衡量 NDN 路由器或网络节点上内容缓存效率。平均请求时延是评估 NDN 网络性能的重要指标之一,反映了用户从发出请求至接收到内容所需的时间。通过测量和分析实际网络中的请求时延,可以更好地了解网络性能并优化网络设计。

3.3 实验结果分析

实验在 Karate 网络拓扑下进行,由于缓存放置策略和缓存替换策略结合使用时能显著提高缓存命中率、降低延迟,并减少冗余量,所以实验中将本文提出的缓存放置策略(CCPS-DRP)以及常见的 LCE、ProbCache、CPCA、NICC 等四种缓存放置策略分别与 FIFO 替换策略结合使用,以对实验结果进行对比分析。

3.3.1 缓存命中率随 Zipf 分布系数的变化

在固定的实验环境下,将 CS 存储容量大小设定为 240(缓存容量比 R 为 0.04),Zipf 分布系数取值范围设定为[0.6,1.8],实验得到五种缓存放置策略分别与 FIFO 替换策略结合使用后对应的缓存命中率随 Zipf 分布系数的变化,如图 3 所示。

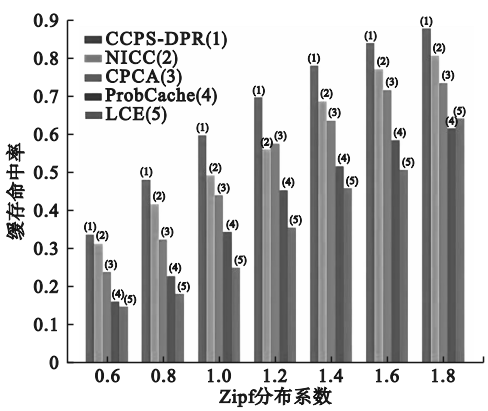

图 3 缓存命中率随 Zipf 分布系数的变化

Fig. 3 The cache hit ratio varied with Zipf distribution coefficient

由图 3 可以看出:随着 Zipf 分布系数增加,各策略的缓存命中率均持续上升;在不同 Zipf 分布系数下,CCPS-DRP 策略的缓存命中率均明显高于其他四种策略;NICC 策略整体优于 CPCA 策略,但 $a=1.2$ 时 NICC 的缓存命中率比 CPCA 落后了 1.4%;ProbCache 策略的缓存命中率介于 NICC 与 LCE 策略之间,但 $a=1.8$ 时较 LCE 低了 2.5%。随着 a 由 0.6 增至 1.8,各策略的缓存命中率增幅情况:CCPS-DRP 达到 51%,NICC 为 49.3%,CPCA 为 49.5%,LCE 为 49.4%,ProbCache 增幅最小,为 45.5%。

3.3.2 平均请求时延随 Zipf 分布系数的变化

在同样的实验环境下,缓存容量比 R 仍设定为 0.04,实验得到五种缓存放置策略对应的平均请求时延随 Zipf 分布系数的变化情况,如图 4 所示。

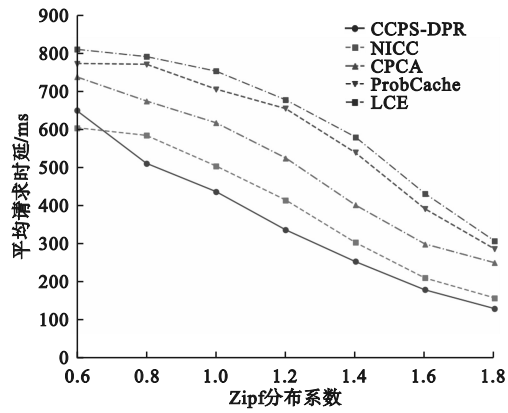

图 4 平均请求时延随 Zipf 分布系数的变化

Fig. 4 The average request delay varied with Zipf distribution coefficient

由图 4 可见:当 Zipf 分布系数为 0.6 时,NICC 策略的平均请求时延最优(比次优的 CCPS-DRP 低 47 ms),当 Zipf 分布系数大于 0.6 时,CCPS-DRP 策略的平均请求时延明显优于其他四种策略;ProbCache 与 LCE 策略的时延整体相近且数值较高;随着 Zipf 分布系数由 0.6 增至 1.8,所有策略的时延均降低且降幅范围在 450~512 ms,其中 CCPS-DRP 策略的平均请求时延降低了 498 ms,表现出全局稳定的优化效果。

3.3.3 缓存命中率随缓存容量比 R 的变化

Zipf 分布系数设定为 0.6,CS 缓存容量设置为 60~480,即缓存容量比为 0.01~0.08,将五种缓存放置策略与 FIFO 替换策略结合,实验得到缓存命中率随缓存容量比 R 的变化情况如图 5 所示。

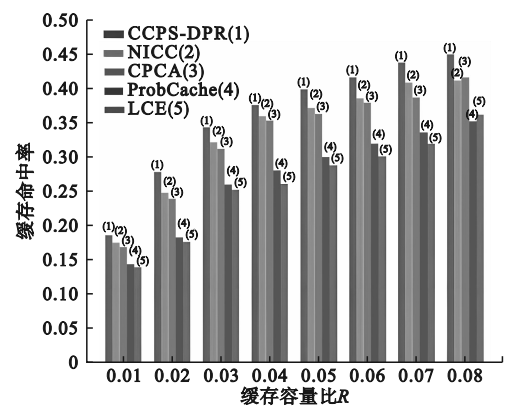

图 5 缓存命中率随 R 值的变化

Fig. 5 The cache hit ratio varied with R-value

由图 5 可见:在不同的缓存容量比下,CCPS-DRP 放置策略的缓存命中率均大于其他四种策略,其中 ProbCache 和 LCE 放置策略的缓存命中率差别不大且数值明显较低,NICC 和 CPCA 策略的缓存命中率接近且数值略低于 CCPS-DRP 策略;随着缓存容量比由 0.01 增至 0.08,五种放置策略的缓存命中率均升高,其中 CCPS-DRP 策略升高了 25.1%,NICC 策略升高了 23.6%,CPCA 策略升高了 24.6%,ProbCache 策略升高了 20.8%,LCE 策略升高了 22.2%。本文提出的 CCPS-DRP 缓存放置策略比其他四种策略在缓存命中率方面具有明显的优势。

3.3.4　平均请求时延随缓存容量比 *R* 的变化

Zipf 系数设定为 0.6,CS 缓存容量为 60 ~ 480,即缓存容量比为 0.01 ~ 0.08,实验得到五种缓存放置策略分别与 FIFO 替换策略相结合使用时平均请求时延随缓存容量比 *R* 的变化情况,如图 6 所示。

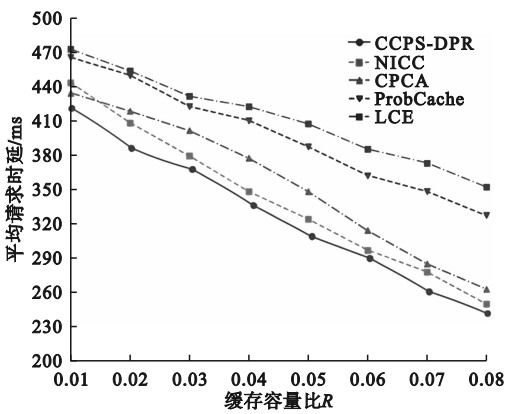

图 6　平均请求时延随 *R* 值的变化

Fig. 6　The average request delay varied with *R*-value

由图 6 可见:在不同的缓存容量比下,CCPS-DRP 缓存放置策略的平均请求时延均低于其他四种策略,其中 LCE 与 ProbCache 策略的平均请求时延比较接近且数值明显较高;随着缓存容量比增大,各策略的平均请求时延均持续下降,但 CCPS-DRP 策略的下降速率最为突出,表明本文提出的缓存放置策略通过优化网络缓存冗余度,有效实现了更大幅度的时延削减,验证了其性能优越性。

3.3.5　缓存命中率随仿真时间的变化

Zipf 分布系数设定为 0.8,CS 缓存容量设置为 360,即缓存容量比 *R* 为 0.06,实验得到五种缓存放置策略分别与 FIFO 替换策略相结合使用时缓存命中率随仿真时间的变化情况,如图 7 所示。

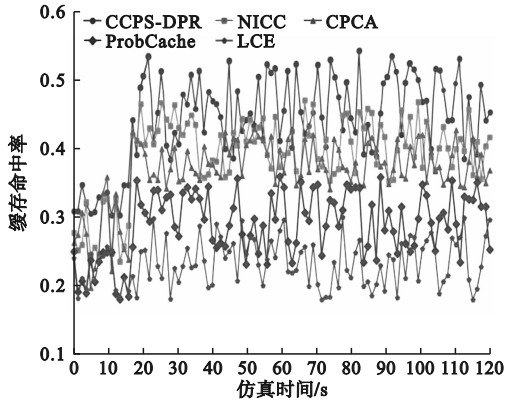

图 7　缓存命中率随仿真时间的变化

Fig. 7　The cache hit ratio varied with the simulation time

由图 7 可见:在初始阶段,CCPS-DRP 策略的缓存命中率与 CPCA、NICC 策略基本持平,且略高于 LCE 与 ProbCache 策略;随着仿真进程推进,CCPS-DRP 策略的缓存命中率明显增大且水平高于其他四种策略。从全局变化趋势来看:LCE 与 ProbCache 策略的缓存命中率曲线波动较大且水平偏低;NICC 与 CPCA 策略的缓存命中率数值演化轨迹比较接近,且保持相对稳定的趋势;CCPS-DRP 策略的缓存命中率始终处于优势区间,表现出较好的性能。

3.3.6　平均请求时延随仿真时间的变化

Zipf 分布系数设定为 0.8,CS 缓存容量设置为 360,即缓存容量比 *R* 为 0.06,实验得到五种缓存放置策略分别与 FIFO 替换策略相结合使用时的平均请求时延随仿真时间的变化情况,如图 8 所示。

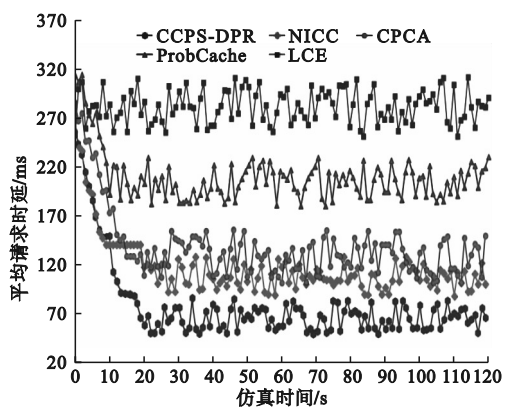

图 8　平均请求时延随仿真时间的变化

Fig. 8　The average request delay varied with the simulation time

由图 8 可见:LCE 策略的时延比较稳定且保持较高水平;ProbCache 策略的时延在前 10 s 内有明显下降,随后稳定在 145 ~ 240 ms;NICC 策略的

时延在前 18 s 呈现急剧下降趋势,之后在 85 ~ 155 ms 范围内震荡;CCPS-DRP 与 CPCA 策略的时延在前 12 s 内具有相似衰减趋势,但 12 s 后 CPCA 策略的时延趋于平缓,而 CCPS-DRP 策略的平均请求时延继续下降,18 s 后趋于平缓,且保持较低的水平。

CCPS-DRP 缓存放置策略与 FIFO 替换策略相结合,展现出较优的时延控制能力,在动态稳定性与持续优化性方面均优于其他四种对比策略,在动态网络环境中具有良好的适应性。

4 结论

为解决多个路由节点缓存相同的内容副本而导致缓存冗余过大、缓存资源未能得到充分利用的问题,本文提出了基于数据冗余包的协同缓存放置策略。采用 GN 策略划分社区,并选择介数中心性高的节点作为关键路由节点,这些节点在网络中承担更多的信息转发任务,适合集中存储高频访问数据。通过集中管理高频数据,避免多个节点重复缓存相同内容,节省了存储空间,且关键路由节点作为数据枢纽,能更快响应邻近节点的请求,减少了数据检索时间。关键节点分担高频数据压力,普通节点仅处理局部请求,避免了节点过载。

通过仿真实验与其他缓存策略对比,充分验证了本策略的优越性,但当前实验基于 Karate 网络,若扩展至大规模网络,CCPS-DRP 放置策略的社区划分与关键节点选择可能面临计算复杂度挑战。今后研究中将考虑融合强化学习与图神经网络,实现动态社区划分与关键节点选择的联合优化,进而更好地扩展至大规模的网络拓扑环境。

参考文献(References):

[1] QU D P,WU J,ZHANG J K,et al. Efficient congestion control scheme based on caching strategy in NDN[J]. Journal of Network and Computer Applications,2023,216:103651.

[2] 任媛,李家兴,张博洋. 浅谈移动边缘网络缓存技术[J]. 科技风,2018(20):89.

[3] GAO J,ZHANG S,ZHAO L,et al. The design of dynamic probabilistic caching with time-varying content popularity[J]. IEEE Transactions on Mobile Computing,20(4):1672 - 1684.

[4] AMADEO M,CAMPOLO C,RUGGERI G,et al. Popularity-aware closeness based caching in NDN edge networks[J]. Sensors,2022,22(9):3460.

[5] 李庆敏,高全力,王西汉,等. 命名数据网络中缓存优化策略的研究[J]. 计算机与数字工程,2022,50(9):1991 - 1997.
LI Q M,GAO Q L,WANG X H,et al. Research on cache optimization strategies in named data networks[J]. Computer & Digital Engineering,2022,50(9):1991 - 1997. (in Chinese)

[6] HEROUALA A T,ZIANI B,KERRACHE C A,et al. CaDaCa:a new caching strategy in NDN using data categorization [J]. Multimedia Systems,2023,29(5):2935 - 2950.

[7] ZHA Y L,CUI P S,HU Y X,et al. An NDN cache-optimization strategy based on dynamic popularity and replacement value[J]. Electronics,2022,11(19):3014.

[8] HOU J C,XIA H Z,LU H Y,et al. A graph neural network approach for caching performance optimization in NDN networks[J]. IEEE Access,2022,10:112657 - 112668.

[9] ALI NAEEM M,BASHIR A K,MENG Y H. Dynamic cluster-based cooperative cache management at the network edges in NDN-based Internet of Things[J]. Alexandria Engineering Journal,2025,125:297 - 310.

[10] ALI NAEEM M,NOR S A,HASSAN S,et al. Compound popular content caching strategy in named data networking [J]. Electronics,2019,8(7):771.

[11] 张子超,郝蔚琳,张伊凡. 一种复杂网络中节点安全重要性排序的度量方法[J]. 信息安全学报,2019,4(1):79 - 88.
ZHANG Z C,HAO W L,ZHANG Y F. A measure approach for ranking the security importance of node security importance in complex network[J]. Journal of Cyber Security,2019,4(1):79 - 88. (in Chinese)

[12] 卢鹏丽,周庚. 基于邻介熵和邻度熵的复杂网络中心性算法[J]. 兰州理工大学学报,2021,47(4):91 - 98.
LU P L,ZHOU G. Centrality algorithm of complex network based on neighborhood betweenness entropy and neighborhood degree entropy[J]. Journal of Lanzhou University of Technology,2021,47(4):91 - 98. (in Chinese)

[13] RESHADINEZHAD A,KHAYYAMBASHI M R,MOVAHEDINIA N. An efficient adaptive cache management scheme for named data networks[J]. Future Generation Computer Systems,2023,148:79 - 92.

[14] WU D,CUI L. A comprehensive survey on segment routing traffic engineering[J]. Digital Communications and Networks,2023,9(4):990 - 1008.

[15] 高子轩,郑烩. NDMANET 中基于内容优先级的缓存策略研究[J]. 计算机工程,2021,47(3):190 - 195.
GAO Z X,ZHENG Q. Research on content priority-based caching strategy in NDMANET[J]. Computer Engineering,2021,47(3):190 - 195. (in Chinese)

[16] QAISER F,HUSSAIN M,AHAD A,et al. Controller-driven vector autoregression model for predicting content popularity in programmable named data networking devices[J]. PeerJ. Computer Science,2024,10:e1854.

[17] DA SILVA E T,DE MACEDO J M H,COSTA A L D. NDN content store and caching policies:performance evaluation [J]. Computers,2022,11(3):37.

[18] HOSSEINZADEH M,MOGHIM N,TAHERI S,et al. A new cache replacement policy in named data network based on FIB table information[J]. Telecommunication Systems,2024,86(3):585 - 596.

[19] YU J T,LENG J C,WU L Y. Network refinement:a unified framework for enhancing signal or removing noise of networks[EB/OL]. arXiv:2109.09119(2021 - 09 - 19)[2025-04-10]. https://doi.org/10.48550/arXiv.2109.09119.

(责任编辑:宋颖韬)