

内容中心网络中一种基于内容等级及流行度的缓存策略

黄胜 刘四军* 滕明隼 姜良浩 郝言明 何玉杰
(重庆邮电大学光通信及网络重点实验室 重庆 400065)

摘要: 网内缓存作为内容中心网络(CCN)的核心技术,越来越受到关注。为了提高网络的缓存性能以及用户对数据请求的满意度,该文提出一种基于内容等级及流行度的缓存策略(CLPC)。CLPC策略通过设置不同用户对内容的访问等级以及统计内容的访问频率,决定数据的缓存位置,并且设计了数据替换方法,在一定程度上保证了不同用户对不同数据的最优请求,同时提高了网络的性能。实验证明该策略与CEE+LRU, Prob(0.7)+LRU, Betw+LRU相比,能更好地提高缓存命中率、平均时延及源端命中率等网络缓存性能指标。

关键词: 内容中心网络; 缓存策略; 内容等级; 流行度

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2017)06-1417-07

DOI: 10.11999/JEIT160816

Cache Strategy Based on Content Level and Popularity in Content Centric Networking

HUANG Sheng LIU Sijun TENG Mingnian JIANG Lianghao HAO Yanming HE Yujie
(Key Laboratory of Optical Communication and Network, Chongqing University of Posts and
Telecommunications, Chongqing 400065, China)

Abstract: As the most important core technology in Content Centric Networking (CCN), caching is widely concerned. In order to improve the network performance and enhance the user satisfaction of data request, a Content-based Level and the Popularity of Caching (CLPC) strategies is proposed. The strategy decides the cache position of data and design method of data replacement by setting the request content level of different users and counting the request frequency, to ensure the optimum request of different users for different data and improve the performance of the network. Experimental results show that compared with the CEE + LRU, Prob (0.7) + LRU and Betw+LRU, the CLPC can increase cache hit ratio, reduce the average delay and the content source hit ratio.

Key words: Content Centric Network (CCN); Caching strategy; Content level; Popularity

1 引言

为了克服 TCP/IP 网络的一些缺陷,研究者提出了内容中心网络(Content Centric Networking, CCN)^[1-3]。CCN 中最重要的是在节点中增加了存储数据的内容存储库(Content Store, CS)^[1],当请求 CS 中已有数据的请求包到达该节点时就可以直接将 CS 中对应的数据返回,无需发往服务器节点,有效地降低了用户请求数据时的时延并减轻服务器

端的压力^[4]。合理的内容放置和缓存替换策略,是提高网络缓存性能的关键^[5]。

目前,国内外研究学者致力于对缓存性能的研究^[6-15],提出了不少的策略。处处缓存策略^[6]将数据存储在数据经过所有节点上,其缺点是:由于网络节点的缓存容量是有限的,导致了网络中内容的多样性降低,进而降低了缓存的利用效率,浪费存储空间。文献[7]提出利用内容转发路径上剩余的缓存容量与已取得的路径收益来计算沿途各路由器缓存该内容的概率,但是该策略没有考虑不同用户对不同数据的请求状况,以及内容的流行度等特征,不能很好地服务用户。基于节点介数策略^[8]的原理是将数据存储在选择介数最大的节点上(最重要的节点),在一定程度上能提高缓存命中率和缓存性能,但是由于将数据存储在选择介数最大的节点,使得内容分布不合理,其他节点的缓存空间得不到有效利用,浪费了存储空间。由文献[14]可知 LRU 策

收稿日期: 2016-08-03; 改回日期: 2016-12-30; 网络出版: 2017-03-07

*通信作者: 刘四军 m13677681001@163.com

基金项目: 国家自然科学基金(61571072, 61371096), 重庆市基础与前沿研究计划项目(cstc2015jcyjA40015, cstc2013jcyjA40052), 重庆市教委科学技术研究项目(KJ130515)

Foundation Items: The National Natural Science Foundation of China (61571072, 61371096), Chongqing Base and Cutting-edge Research Projects (cstc2015jcyjA40015, cstc2013jcyjA40052), Chongqing Commission of Science and Technology Research Projects (KJ130515)

略不能实时反映数据的流行情况, LFU 策略由于其静态特性, 只能盲目地根据数据的请求频率来判断数据是否被替换, 不能及时地反映当前数据请求的流行度, 使当前具有高流行度的数据不能被高效缓存。

针对上述问题, 本文提出了一种基于内容等级及流行度的缓存策略。该策略在保证整个 CCN 网络性能的前提下, 将用户对数据的请求等级作为数据存储的重要依据, 这样能够对重要的数据提供必要的服务, 从而提高用户对数据请求的满意度。并且, 在不影响整个网络请求数据的前提下, 尽可能地将重要数据缓存在距离客户端较近的节点上, 当用户请求数据时, 有效地减少了数据包的传输距离与时延, 同时避免了网络资源的浪费, 一定程度上缓解了网络的拥塞与服务器的压力。

2 CLPC 缓存策略

当前的大部分缓存策略是从网络的角度(路径收益、节点缓存容量等)来提高其缓存性能, 本文基于用户的角度, 提出了一种基于内容等级及流行度的缓存策略, 有如下特点:

(1) 内容等级, 根据不同用户对内容的需求程度划分内容的等级, 提高用户的请求效率以及合理地利用网络资源。

(2) 内容的流行度, 在某段时间内统计某个用户对应的请求内容的等级的访问次数来决定该数据的流行度, 体现数据在网络中的实时性, 增强用户体验, 提高网络性能。

2.1 内容等级

第1节分析了目前CCN中比较有代表性的缓存策略, 大多数策略忽略了用户对数据需求的差异性。由于不同的用户对同一数据往往有不一样的需求, 某数据对某一用户而言可能请求频率较低, 但是, 如果该数据对用户而言很重要(有较高的数据等级), 那么, 我们也需要给该用户提供必要的服务(本文主要是指对数据的缓存和替换)。CLPC 策略依据用户对数据的需求程度来划分数据的等级, 定义等

级越高, 表示客户对该数据需求程度越高, 网络应该保证该数据对该用户的更好服务。

在实际网络中, 用户对数据的需求程度是不同的。因此将用户进行等级的划分在一定程度上保证了等级高的用户能得到更好的服务, 相比于不划分等级, 在一定程度上能减少网络的平均访问时延, 减少网内缓存的冗余, 提高用户的体验。

下面结合图形说明划分等级的优点。

如图1所示, 4个客户端(客户端1,2,3,4)对数据*i*进行请求, 请求等级分别为2,4,3,1, 即客户端2对数据*i*需求最高(等级最高)。在不考虑用户等级的情况下, 数据*i*将在返回的路径上处处缓存, 网络中缓存着大量的内容副本, 导致缓存空间的浪费, 同时忽略了用户对该数据的不同需求。如果只考虑内容的流行度, 不考虑用户的请求等级, 依据流行度的特性, 那么数据*i*将会存储在节点5或者6上, 虽然能服务该链路上的所有用户, 但由于客户端2的请求等级最高, 网络应该优先满足客户端2的请求, 而此时数据缓存的位置相对较远, 导致时延增大, 不能很好地服务重要客户(客户端2), 进而影响整个网络的性能。在考虑用户等级的情况下, 数据会优先存储在靠近等级高的用户节点上, 如图节点1,2, 降低了客户端2,3(等级高的客户端)的访问时延, 实现了对重要客户的优先服务, 从而提高了整体网络的性能。

2.2 内容等级和流行度的计算

在原有CCN模型上, 我们在节点上增加了数据请求信息表(Information Table of Data Request, ITDR), ITDR 中包含了客户端对数据的请求等级及数据在此等级下的请求频率, 用来维护请求在该节点上的数据信息; 另外, 为了使当前节点知道数据*i*在下一跳节点的请求信息, 以判断是否在本节点存储数据, 我们还修改了兴趣包的格式, 加入了数据*i*在本节点上的平均请求等级 q_{ti} 和请求频率总和 F_{ti} 。

根据数据请求信息表ITDR, 可以得到数据*i*在当前节点的请求频率总和 F_{ti} :

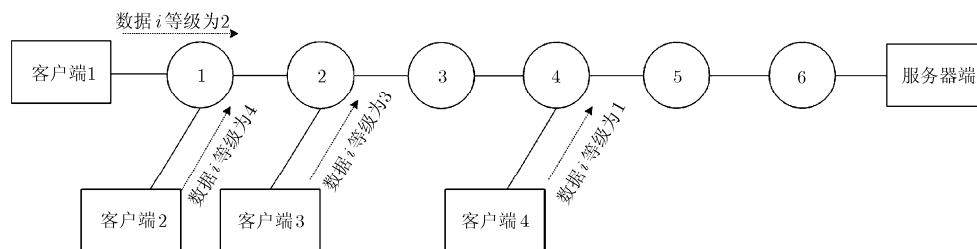


图1 等级划分实例

$$F_{i_i} = \sum_{q_i=\min\{q_i\}}^{\max\{q_i\}} f_{i_{q_i}} \quad (1)$$

其中, F_{i_i} 表示数据 i 在当前节点的请求频率总和, q_i 表示数据 i 的请求等级, $\min\{q_i\}$ 表示数据 i 在节点的最低请求等级, $\max\{q_i\}$ 表示数据 i 在节点的最高请求等级, $f_{i_{q_i}}$ 表示数据 i 在等级为 q_i 时的请求频率。同时, 根据式(1)及 ITDR 可以得出该请求在节点的平均请求等级:

$$q_{i_a} = \left(\sum_{q_i=\min\{q_i\}}^{\max\{q_i\}} q_i f_{i_{q_i}} \right) / F_{i_i} \quad (2)$$

其中, $\sum_{q_i=\min\{q_i\}}^{\max\{q_i\}} q_i f_{i_{q_i}}$ 表示所有用户发送数据 i 的请求在当前节点的等级与相应等级的请求频率积的总和, F_{i_i} 表示数据 i 在当前节点请求频率和。

式(1)和式(2)能够得到客户端对数据在节点的请求情况, 其中, 式(1)体现了数据在节点的流行度(数据在一段时间内不同等级下的频率之和), 式(2)体现了数据在节点的重要性(数据在该节点上相应的平均请求等级), 如果数据在某一节点具有较高的平均请求等级及请求频率, 那么, 说明该节点对此数据很重要, 即很多客户端会到达该节点请求该数据及该数据对请求者而言十分重要, 因此, 可以考虑在此类节点存储该数据。

2.3 CLPC 缓存决策策略

CLPC 决策策略的原理是: 当网络中有数据响应时, 数据包从服务器端发出, 此时, 通过比较该节点 j 与下一跳节点 k 的平均请求等级与请求频率总和和决定数据的放置位置, 如式(3), 式(4)所示:

$$q_{i_{aj}} - q_{i_{ak}} \leq q, \quad q > 0 \quad (3)$$

$$F_{i_{ij}} - F_{i_{ik}} \leq f, \quad f > 0 \quad (4)$$

从式(1)和式(2)可知, $q_{i_{aj}}$ 表示数据在当前节点 j 的平均请求等级, $q_{i_{ak}}$ 表示数据在下一节点 k 的平均请求等级; $F_{i_{ij}}$ 表示数据在当前节点 j 请求频率和, $F_{i_{ik}}$ 表示数据在下一节点 k 请求频率和, q, f 均为大于 0 的常数。

当式(3)和式(4)同时满足时, 即在 t 时刻, 数据 i 在下一跳节点的平均请求等级和请求频率总量都比本节点的高, 则说明下一节点满足数据包缓存条件, 为了使该数据包往靠近客户端的节点缓存, 因此, 将数据包传到下一跳节点, 同时为了避免网内产生过多的冗余数据, 当前节点不存储该数据。这里同时考虑式(3)与式(4)是为了避免数据的请求等级很低或请求频率很低却存储在离客户端较近的节点。分析可知, 通过调整常量 q, f 的值, 可以调整

数据缓存的位置, 若希望数据更靠近客户端节点缓存, 则可以将 q, f 的值适当减小。

2.4 CLPC 缓存替换策略

CLPC 缓存替换策略的原理是: 当数据返回, 按照数据的权重值进行数据的替换。每次需要替换缓存空间中的数据时首先替换掉权重值最小的数据, 如果缓存空间不足时再替换掉权重值次小的数据, 直到数据能够存储在节点上, 若当前数据权重值小于缓存空间中权重值最小的数据, 则无法对缓存空间中已有数据进行替换, 此时, 不缓存该数据。CLPC 策略中, 数据的权重值是依据数据在本节点的客户请求等级及对应的不同时间段内数据的请求频率计算得到的。从 ITDR 表可知, 数据 i 在当前节点的权重值为

$$W_i = \sum_{q=\min\{q_i\}}^{\max\{q_i\}} w_{i_{q_c}} \quad (5)$$

其中, $w_{i_{q_c}}$ 表示数据 i 在等级为 q 时的权值(式 6), W_i 表示数据 i 的总权值。

$$w_{i_{q_c}} = \frac{q_c}{\max\{q_i\}} \left[\theta_1 \sum_{t-\Delta t}^t f_{i_q}(t) + \theta_2 \sum_{t-2\Delta t}^{t-\Delta t} f_{i_q}(t) + \theta_3 \sum_{t-3\Delta t}^{t-2\Delta t} f_{i_q}(t) \right] \sum_{q=\min\{q_i\}}^q \quad (6)$$

其中, q_c 表示当前对数据 i 的请求等级, q 表示数据 i 在当前节点的所有请求等级之和, t 表示是当前时刻, $f_{i_q}(t)$ 表示在某时间段内数据在等级 q 的请求频率, $\theta_i (i \in \{1,2,3\})$ 表示不同时间段对数据 i 的请求频率的比重, 本文取 3 段时间的数据(可以通过调节 Δt 来得到更大的时间段), 即 $\theta_1 + \theta_2 + \theta_3 = 1$, 假设 $\theta_1 > \theta_2 > \theta_3 > 0$, 其目的是为了排除客户端对数据很久前的请求频率很高, 最近的请求频率很低, 而导致总的请求频率较高的情况。通过规定越靠近当前时间段数据的请求频率所占的比重越大, 经过大量仿真, 取值为 $\theta_1 = 0.6, \theta_2 = 0.3, \theta_3 = 0.1$ 时, 本文反映数据的实时性是最好的。

CLPC 策略将数据的平均请求等级和频率总量相结合, 计算出数据在某段时间内的权重值, 根据权重值对数据进行替换, 能更好地反映数据的实时性, 为用户带来更好的访问体验, 提高网络的效率。

2.5 去冗余功能

由图 1 可知, 假设数据 i 已经存储在节点 5 上, 当数据再次请求时, 数据直接从节点 5 返回, 根据 CLPC 策略的原理, 此时节点 3 满足存储条件。如果将数据缓存在节点 3 上, 那么节点 3 和节点 5 都会存储数据, 这样造成了内容副本的冗余, 使得存

储内容的多样性降低, 浪费了网络的存储空间, 降低了网络的缓存利用效率。为了解决内容副本冗余问题, 我们在数据包中添加了数据缓存标识 d , 当数据响应时, 将初值置为 0, 数据包每传到下一跳节点, 就将标识 d 加 1, 通过 d 值可以有效地减少数据在网内的冗余数据, 同时控制整个网络中有一定的数据样本为客户端请求提供响应, 当 d 大于等于阈值 Δd (即 $d \geq \Delta d$) 时 (这里 Δd 大小由网络性能所决定, 其值不能设置太大, 否则会由于数据缓存节点之间过远, 而造成用户在访问该数据时, 网内没有数据满足客户端的请求; 其值不能设置太小, 否则会由于数据缓存节点之间过近, 而造成数据冗余增加以及节点缓存的浪费) 再次由式(3), 式(4)判断当前节点是否能够存储数据, 其余节点以此方式选择出数据放置节点。

2.6 算法描述

在数据包到达节点时, 查看数据包中的数据存储标识 d , 即用来标记该数据所在的当前节点与其最近一次被存储节点的距离, 若标识 d 小于 Δd , 则会将数据传输到下一跳节点, 否则, 通过比较本节点 j 和下一跳节点 k 的该请求客户端的平均请求等级及请求频率, 即由 $q_{iaj} - q_{iak} \leq q$ 和 $F_{tij} - F_{tik} \leq f$ 决定数据是否存储到本节点, 如果两个条件都满足, 则将数据传到下一节点, 否则, 将数据存储到本节点, 同时将数据包中标识 d 置 0, 数据包到下一跳节点后, 标识 d 加 1。

数据决定在某节点存储时, 若此节点缓存空间不足, 依据数据在当前节点请求等级以及该数据在当前节点该等级的不同时间段内的请求频率, 得到反映数据在当前节点请求情况的权重值 W_i , 节点依据 W_i 对数据进行替换并存储。

为了更清楚地说明 CLPC 策略的实现过程, 下面给出节点兴趣包(表 1)和数据包(表 2)处理过程的伪代码。

表 1 Pseudo-code I 兴趣包处理过程伪代码

Initialize ($q_{iaj} = 0, F_{iaj} = 0$)
for each (Router from Content Consumer to Content Server)
Get q_{iak} and F_{tik} from upstream node
Update the ITDR (grade and frequency)
If data in cache
Set $d = 0$
Else Calculate q_{iaj}, F_{tij}

表 2 Pseudo-code II 数据包处理过程伪代码

For each (downstream Router)
If $d \leq \Delta d$
Forward the data to the next hop
$d = d + 1$
Else if $q_{iaj} - q_{iak} \leq q \ \&\& \ F_{tij} - F_{tik} \leq f$
Forward the data to the next hop
$d = d + 1$
Else if The cache buffer is enough
then Cache the data and set $d = 0$
Forward the data to the next hop
else Calculate W_i
compare W_i to W_{\min} (W_{\min} is the minimum value of the data in CS)
If $W_i < W_{\min}$
then $d = d + 1$
Forward the data to the next hop
Else delete the the minimum value of the data in CS
If the cache buffer is enough
then cache the data
$d = d + 1$
Forward the data to the next hop
Else compare W_i to W_{\min} (W_{\min} is the minimum value of the data in CS)

3 仿真与性能分析

3.1 仿真环境和参数设置

本文在 Linux 环境下, 基于 NS-3^[16] 的 ndnSIM^[17] 仿真平台, 通过对 ndnSIM 中各个缓存节点的缓存策略进行配置以及对 CLPC 的自定义编程完成仿真, 并与对比策略进行对比分析。实验采用 ARPA (Advanced Research Project Agency) 网络拓扑对 CLPC 进行仿真, 仿真网络中共有 19 个节点, 其中 4 个客户端分别为 0(c1), 1(c2), 2(c3), 3(c4) 和 1 个服务端 4(root), 其它为中间节点(图 2)。节点请求到达服从泊松分布, $\lambda = 100$ 个/s, 请求概率服从 Zipf 分布。考虑到用户对数据的需求程度以及网络中内容的数量和仿真拓扑, 仿真中为数据等级设置了 4 个等级值(1,2,3,4)。其他主要参数如表 3 所示。

本文采用多个性能参数将 CLPC 策略与 CEE+LRU 策略, Prob(0.7)+LRU 和 Betw+LRU 策略进行比较。下面重点讨论节点缓存大小(cache size)、Zipf(α) 参数及内容数量(content quantity)、数据等级对缓存性能指标的影响。

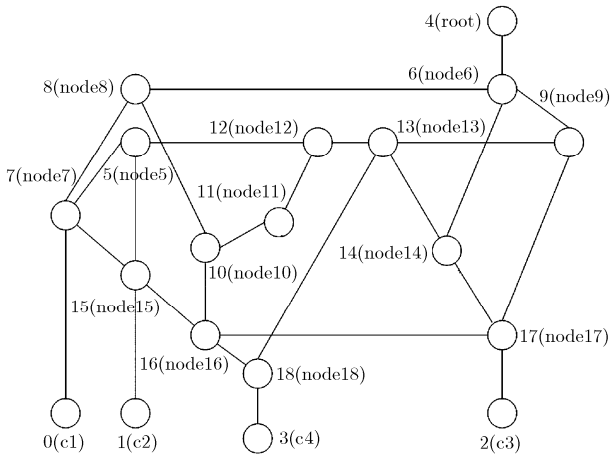


图 2 仿真拓扑

表 3 主要实验参数

主要参数	设值	变化范围
内容数量(个)	1000	200~3200
节点缓存大小(MB)	70	10~190
α 参数	0.7	0.3~1.0
仿真时间(s)	200	

3.2 性能指标

本文的主要性能指标为：(1)缓存命中率 CHR(Cache Hit Ratio)：内容请求在网络中(非原始内容服务器)命中的总数与请求总数的比值。(2)内容源端命中率^[18]CSHR(Content Source Hit Ratio)：内容请求在服务端命中的内容数量与全网命中数量的比值。(3)平均时延 AD(Average Delay)。

3.3 仿真结果分析

(1)节点缓存空间大小的影响：图 3 为 4 种策略随网络节点缓存空间的变化，缓存命中率(CHR)、内容源端命中率(CSHR)、平均时延(AD)变化的情况。从图中可以看出，随着缓存空间的增大，4 种策略的缓存性能都有提高。因为节点缓存空间越大，缓存的内容增多，用户在缓存内容的节点处获取所

需内容的概率增大，从而缓存命中率增大；大量的请求可以在靠近用户端的节点上获取，有效地减轻了服务器的压力，内容源端命中率(CSHR)减小；用户获取所需内容的跳数减少，平均访问时延快速降低。虽然 4 种策略的缓存性能都有明显提升，由于 CLPC 策略考虑了内容的等级和流行度等多方面的因素，使得网内缓存内容多样，内容更加合理地分布在网络节点上，有效地利用网内空间，减轻服务器的压力，相比于 CEE+LRU, Prob(0.7)+LRU 和 Betw+LRU 策略,CLPC 缓存策略的性能更加优秀。当节点缓存大小为 70 MB 时，CLPC 缓存策略比次优的 Betw 缓存策略的缓存命中率(CHR)提高了 35.67%，内容源端命中率(CSHR)降低了 7.49%，平均时延(AD)降低了 26.84%。

(2) Zipf(α) 参数的影响：研究证实网络用户对内容的偏好服从 Zipf 分布^[19]。指数 α 的增大，流行的内容越集中，流行度较高的内容更容易被缓存，并且不容易被替换。图 4 表示，随着指数 α 的增大，CLPC 策略、CEE+LRU, Prob(0.7)+LRU 和 Betw+LRU 等 4 种策略在缓存命中率(CHR)、内容源端命中率(CSHR)、平均时延(AD)等缓存性能指标有所提升。CLPC 策略的性能一直优于其他 3 种策略。主要是因为 CLPC 策略考虑了内容的流行度，能更加反映内容的热度，使得用户对所需内容更容易获取。当节 $\alpha=0.7$ 时，CLPC 缓存策略相比于次优的 Betw 缓存策略的缓存命中率(CHR)提高了 32.79%，内容源端命中率(CSHR)降低了 26.08%，平均时延(AD)降低了 4.63%。

(3)内容数量(content quantity)的影响：图 5 分别为随着内容数量的变化缓存命中率、内容源端命中率和平均时延的变化情况。从图中可以看出，随着内容数量的增大，性能方面都呈现下降的趋势，主要因为随着网内内容数量的增多，在节点缓存大小一定的情况下，需要缓存的内容增多，用户在节点上获取内容的概率降低，导致了缓存性能的下降。CLPC 策略与其他 3 种策略相比，其性能却一直保

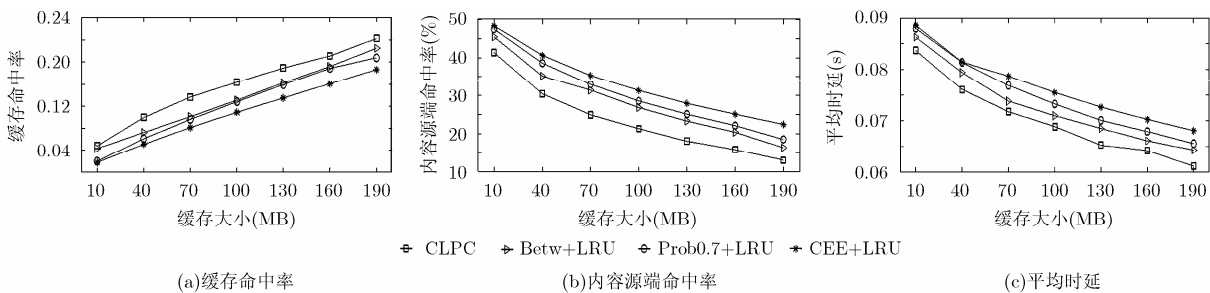


图 3 节点缓存大小对性能的影响

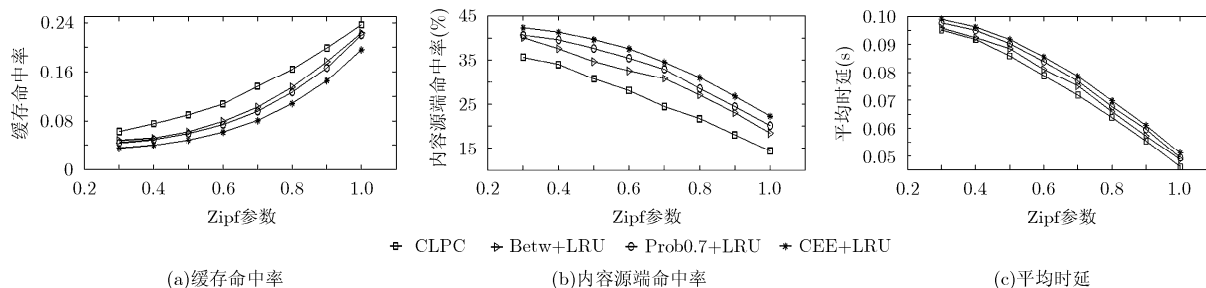
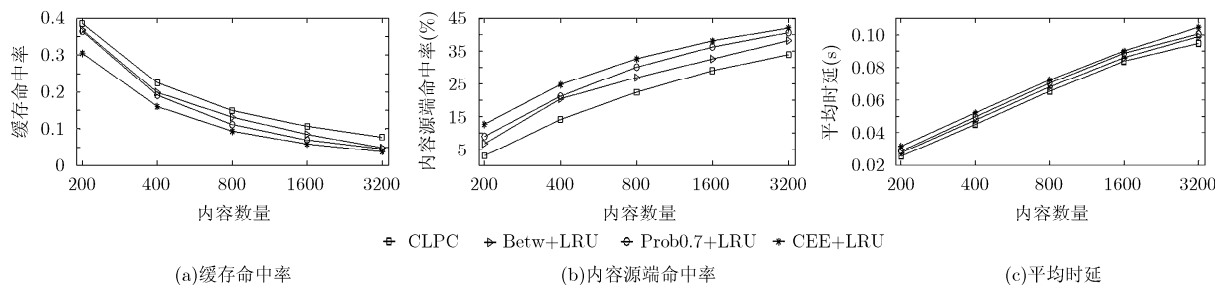
图4 Zipf(α)对缓存性能的影响

图5 内容数量对缓存性能的影响

持着较优的趋势。主要是因为内容数量的增大，相当于节点缓存大小的降低，CLPC策略由于考虑了内容的等级和内容的流行度，使得内容在网络中分布更适合用户的请求，分布更加合理，从而有比较高的性能。当节 $N=800$ 时，CLPC缓存策略相比于次优的Betw缓存策略的缓存命中率(CHR)提高了18.28%，内容源端命中率(CSHR)降低了19.66%，平均时延(AD)降低了3.92%。

(4)请求等级的影响：设置节点缓存大小为70 MB，内容数量为 $N=1000$ 个，Zipf参数为0.7。表4统计了在4种等级下网络的缓存性能。从表中可以看出，随着用户请求等级的增大，缓存性能有所提高。主要是因为随着用户等级增大，CLPC策略中的数据缓存在更靠近客户端的节点上，从而减少用户请求时延，降低了访问跳数，内容更多的存储在网络中的节点而非服务端，减小了源端命中率，同时根据CLPC策略原理可知，对客户端请求等级越高，内容缓存在网络中的可能性越高，从而有效地提高了缓存命中率。

表4 不同等级下的缓存性能对比

数据等级	CHR	CSHR	AD(s)
1	0.11919	0.26223	0.07622
2	0.12263	0.24568	0.07376
3	0.15346	0.20339	0.06848
4	0.22355	0.15365	0.06211

综上所述，本文提出的CLPC缓存策略在缓存命中率(CHR)、内容源端命中率(CSHR)与平均时延(AD)这3种缓存性能参数中都比其他3种缓存策略有更好的缓存性能。

4 结束语

有效地利用CCN网络节点缓存空间，是提高网络性能的关键技术之一。本文提出一种内容等级划分和内容流行度相结合的缓存方法，基于内容等级和访问频率确定内容的缓存位置以及提出相应的缓存替换方法，能够对重要的数据提供必要的服务，从而提高用户对数据请求的满意度，并且，在不影响整个网络请求数据的前提下，尽可能地将重要数据缓存在离客户端较近的节点，当用户请求数据时，有效地减少了数据包的传输距离与时延，同时避免了网络资源的浪费，一定程度上缓解了网络的拥塞与服务器的压力。通过实验仿真，证明了CLPC策略的有效性和高效性。

参考文献

- [1] LEE M, SONG J, CHO K, PACK S, *et al.* Content discovery for information-centric networking[J]. *Computer Networks*, 2015, 83: 1-14. doi: 10.1016/j.comnet.2014.10.006.
- [2] XYLOMENOS G, VERVERIDIS C, SIRIS V, *et al.* A survey of information-centric networking research[J]. *IEEE Communications Surveys and Tutorials*, 2014, 16(2): 1024-1049. doi: 10.1109/SURV.2013.070813.00063.

- [3] LUO H, CHEN Z, CUI J, *et al.* CoLo R: An information-centric Internet architecture for innovations[J]. *IEEE Network Magazine*, 2014, 28(3): 4–10. doi: 10.1109/MNET.2014.6843226.
- [4] 张国强, 李杨, 林涛, 等. 信息中心网络中的内置缓存技术研究[J]. *软件学报*, 2014, 25(1): 154–175. doi: 10.13328/j.cnki.jos.004494.
ZHANG Guoqiang, LI Yang, LIN Tao, *et al.* The built-in caching technology research in Information-Centric Networking[J]. *Journal of Software*, 2014, 25(1): 154–175. doi: 10.13328/j.cnki.jos.004494.
- [5] KYI T, THANT Z O, CHUAN P, *et al.* Efficient forwarding and popularity based caching for content centric network[C]. *IEEE International Conference on Information Networking (ICOIN)*, 2015: 330–335. doi: 10.1109/ICOIN.2015.7057906.
- [6] CHU Weibo, WANG Lifang, XIE Haiyong, *et al.* Network delay guarantee for differentiated services in content-centric networking[J]. *Computer Communications*, 2016, 76: 54–66. doi: 10.1016/j.comcom.2015.09.009.
- [7] PSARAS I, CHAI W K, and PAVLOU G. Probabilistic in-network caching for information-centric networks[C]. *Proceedings of ICN Workshop on Information Centric Networking*, New York, NY, USA, 2012: 55–60. doi: 10.1145/2342488.2342501.
- [8] CHAI W, HE D, PSARAS I, *et al.* Cache “less for more” in information-centric networks (extended version)[J]. *Computer Communications*, 2013, 36(7): 758–770. doi: 10.1016/j.comcom.2013.01.007.
- [9] 刘外喜, 余顺争, 胡晓, 等. CCN 中选择性缓存机制的研究[J]. *计算机学报*, 2014, 37(2): 275–288. doi: 10.3724/SP.J.1016.2014.00275.
LIU Waixi, YU Shunzheng, HU Xiao, *et al.* Selective caching in content-centric networking[J]. *Chinese Journal of Computers*, 2014, 37(2): 275–288. doi: 10.3724/SP.J.1016.2014.00275.
- [10] SHENEELA N, RAO N, AMIR Q, *et al.* Multi-attribute caching: Towards efficient cache management in content-centric networks[C]. *IEEE Annual Consumer Communications & Networking Conference*, Las Vegas, NV, USA, 2016: 630–633. doi: 10.1109/CCNC.2016.7444852.
- [11] WARIT S, WUTTIPONG K, SARAN T, *et al.* Prioritized probabilistic caching algorithm in content centric networks[J]. *Recent Advances in Information and Communication Technology*, 2016, 13(3): 255–265. doi: 10.1007/978-3-319-40415-8_25.
- [12] HU X and GONG J. Opportunistic on-path caching for named data networking[J]. *IEICE Transactions on Communications*, 2014, 97(11): 2360–2367. doi: 10.1587/transcom.E97.B.2360.
- [13] TARNO S, SUPPAKITPAISARN V, and JI Y. Adaptive probabilistic caching for information-centric networking[J]. *IEICE Technical Report*, 2015, 115(210): 63–68.
- [14] CAROFIGLIO G, GALLO M, and MUSCARIELLO L. Bandwidth and storage sharing performance in information centric networking[C]. *Proceedings of the 2011 ACM SIGCOMM Conference*, New York: ACM, 2011: 26–31. doi: 10.1145/2018584.2018593.
- [15] 李俊, 冯宗明, 吴海, 等. 基于层次划分的 CCN 网络缓存存储策略[J]. *通信学报*, 2016, 37(1): 35–41. doi: 10.11959/j.issn.1000-436x.2016005.
LI Jun, FENG Zongming, WU Hai, *et al.* Hierarchical division-based cache storage strategy in content-centric networking[J]. *Journal of Communication*, 2016, 37(1): 35–41. doi: 10.11959/j.issn.1000-436x.2016005.
- [16] HENDERSON T R, ROY S, FLOYD S, *et al.* ns-3 project goals[C]. *Proceeding from the 2006 Workshop on ns-2: The IP Network Simulator*, New York: ACM, NY, USA, 2006: 90–93. doi: 10.1145/1190455.1190468.
- [17] ALEXANDER A, MOISEENKO I, and ZHANG L X. ndnSIM: NDN simulator for NS-3 Named Data Networking (NDN) Project[R]. 2012.
- [18] 崔现东, 刘江, 黄韬, 等. 基于节点介数和替换率的内容中心网络网内缓存策略[J]. *电子与信息学报*, 2014, 36(1): 1–7. doi: 10.3724/SP.J.1146.2013.00503.
CUI Xiangdong, LIU Jiang, HUANG Tao, *et al.* A novel in-network caching scheme based on betweenness and replacement rate in Content Centric Networking[J]. *Journal of Electronic & Information Technology*, 2014, 36(1): 1–7. doi: 10.3724/SP.J.1146.2013.00503.
- [19] BRESLAU L, PEI C, LIN F, *et al.* Web caching and zipf-like distributions: evidence and implications[C]. *IEEE Eighteenth Annual Joint Conference of the IEEE Computer and Communications societies*, California, CA, USA, 1999: 21–25. doi: 10.1109/INFCOM.1999.749260.
- 黄胜: 男, 1974年生, 教授, 研究方向为内容中心网络、信道编码、图像和视频编码。
- 刘四军: 男, 1991年生, 硕士生, 研究方向为内容中心网络缓存策略。
- 滕明埏: 男, 1991年生, 硕士生, 研究方向为命名数据网络数据缓存策略。
- 姜良浩: 男, 1990年生, 硕士生, 研究方向为命名数据网络路由和转发机制。
- 郝言明: 男, 1990年生, 硕士生, 研究方向为命名数据网络中视频数据的传输机制。
- 何玉杰: 男, 1992年生, 硕士生, 研究方向为未来互联网。