



研究与开发

基于内容价值的缓存替换策略

黄丹^{1,2}, 宋荣方^{1,2}

(1. 南京邮电大学通信与信息工程学院, 江苏 南京 210003;
2. 南京邮电大学江苏省通信与网络技术工程研究中心, 江苏 南京 210003)

摘要: 缓存替换机制是内容中心网络的重要研究问题之一, 考虑到缓存空间的有限性, 合理地对缓存内容进行置换, 成为影响网络整体性能的关键因素。因此, 设计了一种基于内容价值的缓存替换方案。该方案综合考虑了内容的动态流行度、缓存代价以及最近被请求的时间, 构建了更实际的内容价值函数, 并依据该内容价值函数, 设计了有效的内容存储与置换方案。具体地, 当缓存空间不足时, 对已有缓存内容按照价值从小到大进行置换。仿真结果表明, 相比于传统替换算法 LRU、LFU 和 FIFO, 本文提出的方案有效地提升了网络节点的内容缓存命中率, 降低了用户获取内容的平均跳数。

关键词: 内容中心网络; 内容价值; 内容替换; 缓存

中图分类号: TN929.5

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2018265

Cache replacement strategy based on content value

HUANG Dan^{1,2}, SONG Rongfang^{1,2}

1. College of Telecommunications & Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
2. Jiangsu Engineering Research Center of Communication and Network Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Abstract: The cache replacement mechanism is one of the important research issues of the content center network. Considering the limited cache space, the reasonable replacement of the cache content becomes a key factor affecting the overall performance of the network. Therefore, a cache replacement scheme based on content value was designed. Considering the dynamic popularity of content, the cost of caching, and the time of the most recently requested, a more realistic content value function was built, and an effective content storage and replacement scheme was designed based on the content value function. Specifically, when the cache space was insufficient, the existing cache contents were replaced according to the value from small to large. Simulation results show that compared with the traditional replacement algorithms LRU, LFU and FIFO, the proposed scheme effectively improves the content cache hit rate of network nodes and reduces the average hops of users' content acquisition.

Key words: content-centric networking, content value, content replacement, caching

收稿日期: 2018-05-08; 修回日期: 2018-09-29

基金项目: 南京邮电大学江苏省通信与网络技术工程研究中心开放课题资助项目; 江苏省自然科学基金资助项目; 南京邮电大学自然科学基金资助项目 (No.217030, No.215161)

Foundation Items: Open Research Fund of Jiangsu Engineering Research Center of Communication and Network Technology, NJUP, Natural Science Foundation of Jiangsu Province, Natural Science Foundation of Nanjing University of Posts and Telecommunications (No.217030, No.215161)



1 引言

随着互联网技术的高速发展,网络信息量和上网用户数剧增,网络流量呈指数级增长,这种爆炸式的流量增长给基于端到端的IP网络体系结构带来了巨大的挑战和压力。为了解决移动网络中巨大的数据流量带来的冲击,研究者们提出了以内容为中心的全新网络架构——内容中心网络(content-centric networking, CCN)^[1]。在内容中心网络中,每一个节点都具有缓存功能,通过将流行内容缓存在距离用户较近的节点上,可以有效地缓解回传链路带宽的压力,降低网络时延,从而改善用户的体验质量^[2]。

但是每个节点的缓存容量是有限的,当缓存空间被占满后,必须根据一定的替换策略将缓存中价值较小的内容清理出来,存放新的有意义的数据,以提高缓存空间的请求命中率,由此可见,合理地对缓存内容进行管理和置换是影响网络整体性能的关键。

传统的缓存替换算法有FIFO(first in first out)、LFU(least frequently used)、LRU(least recently used)以及SIZE替换策略等。其中,FIFO^[3-4]根据队列先进先出的规则,将最先进入缓存空间的数据置换出去,这种算法复杂度低,容易实现,但请求命中率较低。LRU算法^[5]总是将最久未被访问的资源替换出缓存,认为最近被访问的数据在不久的将来被访问的概率也比较高,该算法进行数据置换时仅考虑数据对象最近一次被请求的时间,当内容流行分布发生变化时,适应性能下降。通过统计缓存内容在过去一段时间内的访问频率,LFU替换策略^[6]认为频率越高的资源,使用价值就越高,因此当缓存空间不足时,总是替换访问频率最低的内容,该算法也存在缓存污染问题,即过去访问频率高的内容即使现在不再被访问依然占据缓存空间,致使缓存空间的利用率降低。SIZE算法^[7]根据内容的大小置换数

据,优先将字节数大的数据替换出去,但这种算法没有考虑访问时间间隔、缓存命中次数等因素,可能会使价值不高的小字节对象长时间保留在缓存空间中,从而降低命中率。

以上算法的设计略显片面,采用单一目标函数来确定缓存替换对象已经不能满足多样化的网络数据需求。参考文献[8]将LRU和LFU算法结合提出了一种新的LFRU策略,该算法综合考虑了内容的请求频率和最近访问的时间,但是未考虑内容的全局流行度。参考文献[9]提出的GDS(greedy-dual size)算法综合考虑了对象大小、缓存代价以及内容的年龄因子,给缓存空间的每个文件设置权重,每次替换权重最小的对象,但是该算法未考虑缓存对象过去被访问的次数。参考文献[10]提出了一种改进的WGDSF缓存替换策略,该策略在GDS算法的基础上加入了基于时间的加权频率参数和加权文件类型参数,虽然性能有所提升,但是也极大地增加了算法复杂度。参考文献[11]提出了一种LUV-Path策略,该策略将内容的价值设为其所在节点与源服务器之间的距离,距离越大,内容的价值越高,该策略充分考虑了内容的请求代价,但未考虑内容的流行度,不能准确反映数据当前的流行程度。参考文献[12]则提出一种基于内容流行度的缓存替换策略,该策略根据数据的命中率、最近请求时间间隔以及请求次数等因素确定数据块的流行度,每次替换流行度最低的数据。参考文献[13]提出了一种最小化内容中心网络中传输开销的协作缓存机制,根据内容被请求的次数以及网络中不同域之间传输数据的单位链路开销确定内容在某个域中的价值。参考文献[14]提出了一种基于PageRank的替换算法,该策略将用户和文件看作网络中的节点,当用户访问了数据对象后,用户和该文件的关系被抽象为网络中的连线,然后根据PageRank算法计算出用户和数据内容的权重值,每次优先替换权重值最小的数据对象。该算法主要是从用户的

访问行为方面考虑,而忽略了内容本身的特性。

针对以上问题,本文设计了一种基于内容价值的缓存替换策略,该算法综合考虑了内容的动态流行度、缓存代价以及缓存内容最近一次被请求的时间来确定缓存对象的存储价值,当缓存空间不足时,将价值最小的数据置换出去。仿真实验结果表明,该算法能够有效地提高缓存命中率,减少用户获取数据的平均跳数。

2 CCN 工作原理

CCN 是一种全新的通信网络架构,该架构不再关心内容存放的位置,而只关注数据本身,结构中的数据报文以内容名称作为标识而不是传统的 IP 地址,CCN 中有两种数据分组类型:兴趣分组(interest packet)和数据分组(data packet)^[15]。兴趣分组由请求客户发出,其中携带请求内容的名字,但不分组含目的地址,兴趣分组被路由到有请求资源的邻近节点或源服务器后,产生相对应的数据分组,数据分组中携带数据对象、内容名字以及发布者的签名信息,它会沿着兴趣分组的反向路径传送给用户^[16]。

CCN 路由节点主要由内容存储器(content store, CS)、转发信息库(forwarding information base, FIB)和未决请求表(pending interest table, PIT)3 个基本数据表组成^[17]。CS 存储到达该节点上的数据对象,缓存下来的内容可以用来满足未来对该数据的请求,从而减少内容的重复传输。FIB 类似传统的 TCP/IP 网络架构中的 IP 路由表,但目标字段是内容名称的前缀。PIT 记录正在传输的路由状态信息。当节点上有兴趣分组到达时,首先会查询 CS 中是否存储了该兴趣分组对应的内容,如果有就将内容沿着兴趣分组的到达路径直接发送给请求者,同时丢弃该兴趣分组。如果缓存中没有该内容,节点会继续查询 PIT 中是否有与内容名对应的条目,如果存在就将请求接口添加到对应的接口列表中,然后将该请求兴趣分

组丢弃,否则创建一条新的 PIT 条目并查询 FIB,将兴趣分组按照匹配的端口转发到邻近的 CCN 节点,若 FIB 中没有匹配的条目,则将兴趣分组丢弃或转发到默认的端口。

当数据分组到达时,节点会查询 PIT 中是否存在与该内容对应的条目,若存在则将数据内容从条目的接口列表中转发出去,并删除对应的 PIT 表项,然后根据相应的缓存替换策略将该数据存储于 CS 中。

3 基于内容价值的缓存替换策略

CCN 网络中的每个路由器节点都具有缓存能力,缓存的数据内容能满足这些数据对应的后续请求。但是节点的存储容量有限,随着用户请求数量的累积以及用户访问内容的动态变化性,需要对 CS 中存储的对象进行替换,高效的缓存置换策略可以提高网内数据请求在边缘节点的命中率,减少内容的重复传输,降低服务器端的负载压力。基于以上分析,本文提出了一种基于内容价值的缓存替换策略,综合考虑了数据内容的流行度分布、缓存代价以及内容最近被请求的时间。该策略不仅实时地更新内容的流行度,能够将用户受欢迎的数据对象及时地存储下来,而且对缓存代价较大的数据也能很好地缓存,在提高命中率的同时还能极大地减少网内的流量开销,降低用户获取数据的平均跳数,改善用户的上网体验。

3.1 内容流行度

在 CCN 中,内容的流行度能够很好地反映用户对内容的需求情况。由于用户对数据请求的时变性,网络中文件内容的流行度总是不停地发生变化,例如缓存中的某些内容在当前时间段内受到用户的高度关注,流行度很高,但是随着时间的流逝,这部分内容很可能变成用户较少访问的对象,流行度下降。因此,受用户访问行为的影响,网内节点上存储的内容的流行度是动态变化的,即需要在每个周期内对内容的流行度进行统



计更新。为了准确预测其变化,本文在每个路由器节点中,为缓存空间中的每项内容设计一个计数器,用来统计每个周期内该内容被命中的次数,同时设计一个总计数器来统计节点收到的总请求数。然后基于指数加权移动平均算法(EWMA)^[18],根据上个周期内容的流行度以及当前周期内该内容的命中率来计算当前周期的内容流行度:

$$p_r(t) = \alpha p_r(t-1) + (1-\alpha)h_r(t) \quad (1)$$

$$h_r(t) = \frac{N_r(t)}{N_Q(t)} \quad (2)$$

其中, $p_r(t)$ 表示内容 r 在当前周期内的流行度, $p_r(t-1)$ 表示内容 r 在上一个周期内的流行度, α 是衰减因子, $0 < \alpha < 1$, 表示上一个周期的流行度在当前周期所占的比例, $h_r(t)$ 表示内容 r 在当前周期内的命中率, $N_r(t)$ 表示当前周期内内容 r 被命中的次数, $N_Q(t)$ 表示当前周期内路由节点收到的总请求数。

3.2 缓存代价

由于 CCN 每个节点中的 CS 都能存储多个数据对象,同一 CS 存储这些数据内容所消耗的网络资源不同,因此在进行数据替换时要充分考虑 CS 中内容的缓存代价。如果将高缓存代价的数据对象进行替换,那么,当下次该数据的请求到达节点时,节点将不能响应该请求,只能将此数据请求转发到其他节点或源服务器以获取内容。这样用户获取该数据会消耗更多的网络带宽资源。

本文中缓存代价主要由传输代价和缓存成本两部分组成,传输代价主要考虑请求内容所需的跳数。因此本文在数据分组中添加了一个字段用来记录数据分组从源服务器到达命中节点经过的跳数,跳数越大,传输代价越高,内容的存储价值就越大,因此在缓存替换过程中应尽量避免将距源服务器较远的内容替换掉。

综上所述,CS 中内容的缓存代价如式(3)所示:

$$C_r = \text{Hop}_r \times p_r^t + p_r^s \quad (3)$$

其中, Hop_r 表示内容 r 距离源服务器的跳数, p_r^t 表示内容 r 单跳的传输成本, p_r^s 表示内容 r 的缓存成本。

内容的传输代价远远大于其缓存成本,因此为了方便后续处理,令:

$$\eta = \frac{\text{Hop}_r \times p_r^t}{p_r^s} \quad (4)$$

$$C_r = \frac{\eta+1}{\eta} \times p_r^t \times \text{Hop}_r \quad (5)$$

其中, $\eta \gg 1$ 。

3.3 数据内容的缓存价值计算

本文在 LRU 的基础上为缓存节点中的每项内容设计了一个时间标签,用来记录该内容最近一次被访问的时间,假设当前时间为 t_{cur} ,内容最近一次请求的时间为 t_{old} ,则:

$$t_{\text{inter}} = t_{\text{cur}} - t_{\text{old}} \quad (6)$$

时间间隔 t_{inter} 越短,说明该内容在当前时间段内被请求的频率越高,内容的存储价值主要取决于内容的流行度,而当间隔 t_{inter} 越长时,影响内容价值的主要因素就是内容的缓存代价,内容的流行度作为次要因素。基于以上分析,缓存内容的价值函数如下所示:

$$\text{value}_r(t) = \frac{1}{e^{t_{\text{inter}}}} p_r(t) + (1 - \frac{1}{e^{t_{\text{inter}}}}) C_r \quad (7)$$

由式(7)可以看出,如果短时间内该数据被频繁请求,那么时间间隔 t_{inter} 很小,用户对该内容的关注度很高,说明此时该数据对象的流行度很高,应将其作为影响缓存内容价值的主要因素,反之当间隔 t_{inter} 很大时,说明该内容最近被请求的次数较少,内容的流行度下降,此时应将内容的缓存代价作为影响价值函数的主要因素,将内容流行度作为次要影响因素。该策略利用内容被请求的时间间隔作为权重因子,可以实时调整数据的流行度和缓存代价在价值函数中所占的比重,从而不仅能将高流行度的内容缓存在路由节点中,而且对高请求代价的数据也有很好的响应,

使网络中存储的数据更加合理。

3.4 算法实现

缓存空间中的每项内容都有对应的存储价值,当缓存容量不够时,就将价值最小的内容替换出去,以满足新内容的存储要求,兴趣分组处理过程的伪代码如下所示:

```

输入: 内容兴趣分组  $r$ 
if CS find(content  $r$ )
    增加请求内容  $r$  的命中次数和节点的总请求数
    更新内容  $r$  的请求时间
    准备将内容  $r$  发送给用户
else
    if PIT find (content  $r$ )
        丢弃兴趣分组并更新 PIT
    else
        在 PIT 中添加兴趣分组
        根据 FIB 转发兴趣分组
    end if
end if
数据分组处理流程的伪代码如下所示:
输入: 内容兴趣分组  $r$ 
if CS find(content  $r$ )
    丢弃数据分组
else
    if PITfind(content  $r$ )
        根据 PIT 向用户发送内容  $r$ 
    if CS 有足够空间
        CS save(content  $r$ )
        初始化内容  $r$  的命中数和请求时间
        记录数据分组到达该节点经过的跳数
    else
        根据当前时段的内容流行度、缓存成本和请求时间间隔计算 CS 中缓存的每个内容的值
        删除最不重要的内容项

```

end if

end if

end if

4 仿真与分析

4.1 实验设计

为了评估本文提出的替换算法的可行性和对 CCN 性能的提高程度,本文设计了仿真实验进行验证。假设网络中的文件总数默认为 5 000 个,变化范围是 2 000~8 000,每个文件的大小都相同,内容数据分组的流行度服从 Zipf-Mandelbrot 分布^[19],默认参数 $\beta=1.0$,变化范围是 0.7~1.3, $k=80$,请求频率服从参数 $\lambda=100$ 请求/s 的泊松分布,仿真时间为 100 s,流行度的更新周期为 5 s,衰减因子 α 为 0.4,为了便于仿真实验的进行,假设每个路由器节点的缓存容量相同,缓存容量默认为数据总数的 8%,变化范围是 2%~14%。存储模式选用 LCE (leave copy everywhere),将本文的算法分别与传统的替换策略 LRU、LFU、FIFO 进行比较分析。

4.2 性能指标

为了客观地衡量不同缓存替换算法的实际性能以及不同参数对缓存系统的影响,本文研究主要采用以下两个评价指标,分别为缓存节点的平均命中率和用户获取数据对象的平均路由跳数比。

(1) 缓存节点的平均命中率

即在一定的时间段内,用户发出的总请求数为 N_{req} ,数据请求在缓存节点内命中的次数为 N_{hit} ,平均命中率 $r = \frac{N_{\text{hit}}}{N_{\text{req}}}$, r 越大,缓存效果越佳。

(2) 平均路由跳数比

即在一定的时间段内,用户发出的数据请求在缓存节点命中时经过的路由跳数 hop_c 与用户在源服务器获取内容经过的跳数 hop_s 之间的比值

$$\eta = \frac{\sum_{N_{\text{req}}} \text{hop}_c}{\sum_{N_{\text{req}}} \text{hop}_s}, \eta \text{ 越小,越能减少网内的流量开销,}$$

用户的上网体验越佳。



4.3 实验结果分析

4.3.1 Zipf-Mandelbrot 分布参数 β 的影响

目前, 内容的流行度服从 Zipf-Mandelbrot 分布, 研究者发现在不同的场景下 Zipf-Mandelbrot 分布参数的值具有差异性, 因此本文通过改变 Zipf-Mandelbrot 分布参数 β 来研究其对缓存系统的影响, 同时也比较不同缓存替换算法的性能。

图 1 和图 2 分别展示了缓存节点的平均命中率和用户获取内容的平均跳数比随着分布参数 β 变化的实验结果。从图 1 中可看出, 当分布参数 β 较小时, 由于流行度高的内容分布范围较广, 流行度高的内容和流行度低的内容区分不明显, 所以网内节点缓存的数据对象包括了各个分布区域的内容, FIFO、LRU 和 LFU 缓存替换策略无法准确预测内容流行度的变化, 因此大部分的数据请求只能从源服务器端获得, 缓存节点的命中率较低。随着参数 β 逐渐增大, 流行度高的内容分布较集中, 用户请求的内容更加具有局部性, 更有利于缓存系统充分发挥其存储价值, 各种缓存替换策略的性能都有所改善。同时, 实验结果表明, 本文提出的基于内容价值的缓存替换策略相比于其他传统的替换算法在参数 β 较小时有一定优势, β 越大, 优势越明显, 说明该算法使缓存系统的性能得到了进一步的提升。同样在图 2 中, 由对图 1 的分析可知, 随着参数 β 的增大, 网内节点的命中率提升, 用户获取内容的跳数则逐渐降低。

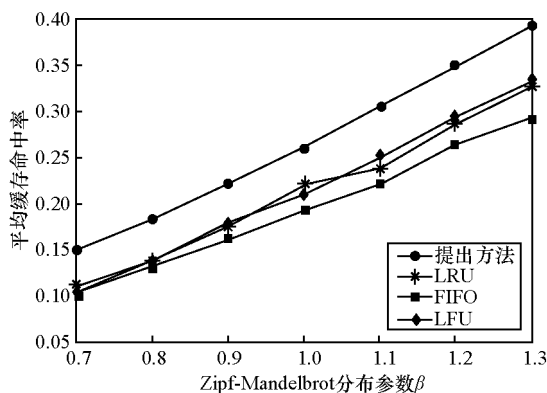


图 1 平均缓存命中率

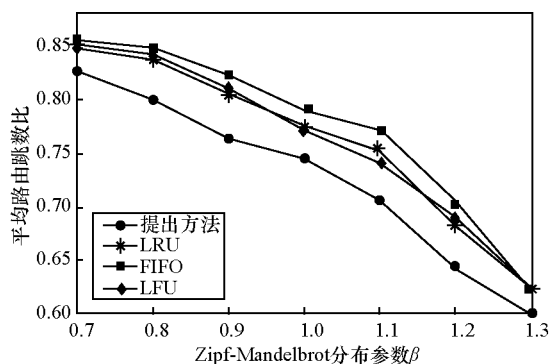


图 2 平均路由跳数比

4.3.2 网内节点缓存容量的影响

通过改变节点缓存容量的大小来研究缓存性能的变化。图 3 和图 4 分别为网内节点的平均命中率和获取内容的平均跳数比随着网内缓存容量变化的示意图。从图 3 中可以看出, 当缓存容量从 2% 增加到 14% 时, 各个缓存替换策略的平均命中率都有所提高, 说明增大缓存容量可以改善系统性能, 但同时也发现随着缓存容量不断增大, 平均缓存命中率的曲线上凸, 说明性能改善的速率下降。此外, 从图 3 中可以发现本文提出的基于内容价值的缓存替换算法比其他 3 种缓存策略的性能都好, 尤其当缓存容量较小时, 这种优势更加明显。图 4 表明当网内节点的缓存容量增大时, 每种替换策略的平均请求跳数都呈下降趋势。相比于其他几种算法, 本文的替换策略能够大大降低用户获取内容的平均跳数, 提高网络的传输效率。该策略将传输代价作为影响价值函数的一个重要因素, 避免将距离内容源服务器较远的内容替换掉, 同时也充分考虑了用户的需求, 将流行度较高的内容缓存在边缘节点中, 因此该算法在一定程度上减少了用户获取内容的平均跳数。

4.3.3 内容总量的影响

本文通过改变内容总量的大小来研究系统性能的变化, 图 5 显示了数据请求在网内节点的命中率。从图 5 中可以看出, 当内容总量从 2 000 增加到 8 000 时, 缓存节点的命中率是呈下降趋势的, 这是因为当其他参数不变时, 内容总量的增

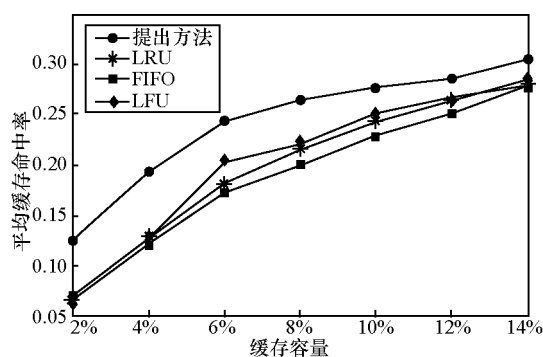


图3 平均缓存命中率

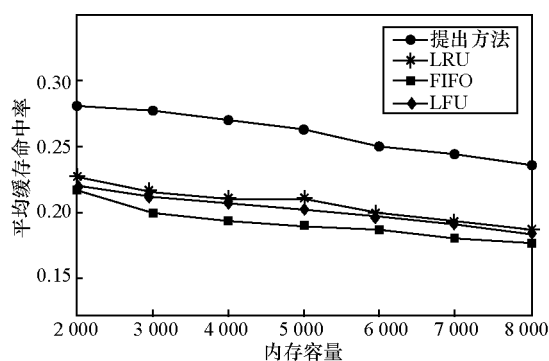


图5 平均缓存命中率

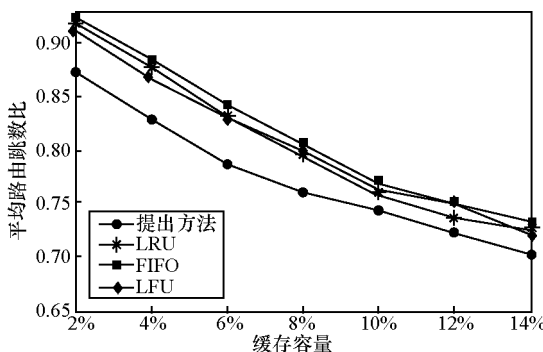


图4 平均路由跳数比

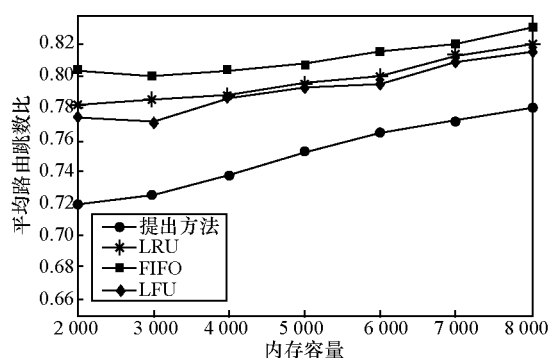


图6 平均路由跳数比

大使内容的局部性相对减弱，用户的数据请求更加分散，导致系统的缓存性能下降。但是本文提出的基于内容价值的缓存替换策略与 FIFO、LRU 和 LFU 相比在网内节点的命中率要高，主要归因于该算法动态更新内容的流行度，用内容的请求时间差作为调节因子，使得缓存空间的数据对象更能满足用户的后续请求。

图 6 是用户请求内容的平均路由跳数比随内容总量变化的示意图。由图 5 的分析可知，随着内容总量的增大，用户的数据请求在网内节点的命中率降低，因而获取内容的平均跳数就上升。由图 6 可知，本文的替换策略与传统缓存替换算法相比，用户获取数据的平均路由跳数比要低。

5 结束语

为了提高 CCN 内容利用的效率，本文提出了一种新的基于内容价值的缓存替换策略，该算法综合考虑了数据内容的动态流行度、缓存代价和内容请求时间差，将内容的请求时间间隔作为调节

因子，实时调整内容流行度和缓存代价所占的比重。实验结果表明，该策略能够显著地提高缓存节点的命中率，降低用户获取内容的平均跳数，从而改善网络的整体性能。

本文中将缓存节点看作一个独立的个体，下一步的工作将考虑缓存节点间的协作关系，实现协作式缓存管理，可以减少节点间缓存内容的冗余度，进一步提高 CCN 内缓存内容的多样性。

参考文献:

- [1] AMADEO M, CAMPOLO C, MOLINARO A, et al. Content-centric wireless networking: a survey[J]. Computer Networks, 2014, 72(7): 1-13.
- [2] 张建敏, 谢伟良, 杨峰义, 等. 移动边缘计算技术及其本地分流方案[J]. 电信科学, 2016, 32(7): 132-139.
ZHANG J M, XIE W L, YANG F Y, et al. Mobile edge computing and application in traffic offloading[J]. Telecommunications Science, 2016, 32(7): 132-139.
- [3] MUSCARIELLO L, CAROFIGLIO G, GALLO M. Bandwidth and storage sharing performance in information centric networking[C]//ACM SIGCOMM Workshop on Information-Centric Networking, Aug 15-19, 2011, Toronto, Canada.



- New York: ACM Press, 2011: 26-31.
- [4] 董利利, 王勇, 董永强, 等. NDN 中基于蚁群替换算法的邻居协作缓存管理策略[J]. 电信科学, 2014, 30(9): 45-52.
DONG L L, WANG Y, DONG Y Q, et al. Collaborative cache management strategy based on ant-colony replacement algorithm in named data networking[J]. Telecommunications Science, 2014, 30(9): 45-52.
- [5] GUO J, LIU C, CHEN J, et al. S-LRU: a cache replacement algorithm of video sharing system for mobile devices[C]//International Conference on Computer Science and Network Technology, Dec 29-31, 2012, Changchun, China. Piscataway: IEEE Press, 2012: 1180-1184.
- [6] ZHU X, YANG Y, LIU Z, et al. C-aware: a cache management algorithm considering cache media access characteristic in cloud computing[J]. Mathematical Problems in Engineering, 2013(6): 1-13.
- [7] MA T H, TIAN W, WANG B, et al. Weather data sharing system: an agent-based distributed data management[J]. IET Software, 2011, 5(1): 21-31.
- [8] DAS S, BANERJEE A. An arbitration on cache replacements based on frequency—recency product values[C]//International Conference on VLSI Systems, Architectures, Technology and Applications, Jan 10-12, 2016, Bangalore, India. Piscataway: IEEE Press, 2016: 1-6.
- [9] CAO P, IRANI S. Cost-aware WWW proxy caching algorithms[C]//USENIX Symposium on Internet Technologies and Systems, Dec 8-11, 1997, California, USA. Berkeley: USENIX Association, 1997: 18.
- [10] MA T, QU J, SHEN W, et al. Weighted greedy dual size frequency based caching replacement algorithm[J]. IEEE Access, 2018, PP(99): 1.
- [11] CHEN X, FAN Q, YIN H. Caching in information-centric networking: from a content delivery path perspective[C]// International Conference on Innovations in Information Technology, Mar 17-19, 2013, Abu Dhabi, United Arab Emirates. Piscataway: IEEE Press, 2013: 48-53.
- [12] CHOOTONG S, THAENTHONG J. Cache replacement mechanism with content popularity for vehicular content-centric networks (VCCN)[C]//International Joint Conference on Computer Science and Software Engineering, Jul 12-14, 2017, Nakhon Si Thammarat, Thailand. Piscataway: IEEE Press, 2017: 1-6.
- [13] 霍跃华, 刘银龙. 内容中心网络中传输开销最小的协作缓存策略[J]. 太原理工大学学报, 2017, 48(1): 116-121.
- HUO Y H, LIU Y L. Collaborative caching strategy based on minimizing transmission in content-centric networking[J]. Journal of Taiyuan University of Technology, 2017, 48(1): 116-121.
- [14] 肖敬伟, 赵永祥. 基于 PageRank 的缓存替换策略[J]. 信息技术, 2016(6): 107-110.
XIAO J W, ZHAO Y X. Cache replacement strategy based on PageRank[J]. Information Technology, 2016(6): 107-110.
- [15] XYLOMENOS G, VERVERIDIS C N, SIRIS V A, et al. A survey of information-centric networking research[J]. IEEE Communications Surveys & Tutorials, 2014, 16(2): 1024-1049.
- [16] WU H, LI J, PAN T, et al. A novel caching scheme for the backbone of named data networking[C]//IEEE International Conference on Communications, Jun 9-13, 2013, Budapest, Hungary. Piscataway: IEEE Press, 2013: 3634-3638.
- [17] ZHANG M, LUO H, ZHANG H. A survey of caching mechanisms in information-centric networking[J]. IEEE Communications Surveys & Tutorials, 2015, 17(3): 1473-1499.
- [18] LING Q, XU L, YAN J, et al. An adaptive caching algorithm suitable for time-varying user accesses in VOD systems[J]. Multimedia Tools & Applications, 2015, 74(24): 11117-11137.
- [19] BRESLAU L, CAO P, FAN L, et al. Web caching and Zipf-like distributions: evidence and implications[C]//IEEE Joint Conf Comput Commun Soc (INFOCOM), Mar 21-25, 1999, New York, USA. Piscataway: IEEE Press, 1999: 126-134.

[作者简介]



黄丹 (1993—), 女, 南京邮电大学通信与信息工程学院硕士生, 主要研究方向为内容中心网络中的缓存机制。



宋荣方 (1964—), 男, 南京邮电大学通信与信息工程学院教授、博士生导师, 主要研究方向为宽带无线通信理论与技术。