

内容中心网络中基于缓存压力的选择性缓存机制^{*}

罗熹^{1,2}, 安莹^{1†}

(1. 中南大学 信息安全与大数据研究院, 湖南 长沙 410083; 2. 湖南警察学院 信息技术系, 湖南 长沙 410138)

摘要: 内置缓存机制是内容中心网络(Content Centric Networking, CCN)研究的关键问题之一。现有的CCN缓存机制在缓存节点的选择时往往缺乏对节点缓存状态和负载情况等动态信息的综合考虑,因此无法准确反映节点间的缓存状态差异,进而影响了缓存决策的合理性。提出了一种基于缓存压力的选择性缓存机制(Cache Pressure Based Selective Caching, CPBSC)。该机制利用节点的缓存占用率和缓存替换率评估节点的缓存压力,并结合内容的流行度实现缓存节点的合理选择。同时,根据内容的请求率和兴趣分组的剩余生存时间评估不同内容的缓存价值,以此实现缓存内容的合理替换。仿真结果表明,CPBSC有效地优化了缓存资源利用率,并显著提高了系统的整体缓存性能。

关键词: 下一代网络; 内容中心网络; 缓存压力; 内置缓存; 选择性缓存

中图分类号: TP393

文献标志码: A

Cache Pressure Based Selective Caching Scheme for Content-centric Networks

LUO Xi^{1,2}, AN Ying^{1†}

(1. Institute of Information Security & Big Data, Central South University, Changsha 410083, China;

2. Department of Information Technology, Hunan Police Academy, Changsha 410138, China)

Abstract: In-network caching scheme is one of the key technologies of Content Centric Networking (CCN). In some existing CCN caching schemes, the dynamic information of cache state and load condition are not fully considered so that the differences of cache state between nodes cannot be reflected accurately, which affects the reasonability of caching decision. In this paper, a Cache Pressure Based Selective Caching Scheme (CPBSC) was proposed. In this scheme, the cache occupancy rate and cache replacement rate were used to estimate the cache pressure of node. According to the cache pressure and content popularity, the caching nodes were rationally selected. Meanwhile, the content replacement is implemented based on the cache value of content which is evaluated by the content request rate and the residual time to live of Interest packet. Simulation results show that CPBSC effectively increases the resource utilization and improves the whole cache performance.

^{*} 收稿日期:2017-05-08

基金项目: 国家自然科学基金资助项目(61402541), National Natural Science Foundation of China(61402541); 湖南省科技计划重大专项(2017SK1040), Hunan Province Science and Technology Key Projects(2017SK1040); 网络犯罪侦查湖南省普通高等学校重点实验室开放研究基金资助项目(2016WLFZZC009), Open Research Fund of Key Laboratory of Hunan Normal University for Cyber Crime Investigation(2016WLFZZC009); 网络侦查技术湖南省重点实验室开放研究基金资助项目(2016WLZC009), Open Research Fund of Hunan Province for Network Investigation Technology(2016WLZC009)

作者简介: 罗熹(1980-), 女, 湖南长沙人, 湖南警察学院讲师, 博士

[†] 通讯联系人, E-mail: anying@csu.edu.cn

Key words: next generation networks; Content-Centric Networking; cache pressure; in-network caching; selective caching

随着网络通信技术的发展,互联网在提供的服务种类以及交换的数据量等方面均出现了迅速的增长.网络应用的主体已从最初的计算资源共享逐步向内容的产生、分发以及获取服务转变.传统的以主机为中心(Host-centric)的网络体系结构已经无法适应当前互联网发展的需要.为了满足未来不断增长的内容访问需求,学术界展开了对未来互联网体系架构的广泛探索,其中,内容中心网络(Content-Centric Networking, CCN)^[1]作为最具代表性的解决方案受到了极大的关注.它将内容本身作为网络通信的主体,实现了从以主机为中心(Host-centric)的 Where 会话方式向以内容为中心(Content-centric)的 What 会话方式的演进.

CCN 架构中,用户通过发送兴趣分组(Interest packet)请求自己需要的内容.当网络中的节点收到兴趣分组时,将根据内容的名称进行匹配查询.若发现与兴趣分组匹配的内容,则沿着该兴趣分组的反向传播路径返回相应的内容数据分组(Data packet).同时,CCN 采用泛在化的网络内置缓存(In-network Caching)方式^[2],在返回内容数据分组的沿途转发路径上对应答内容进行缓存,使得网络作为内容传输体的同时也成为了内容的存储体.内置缓存机制使得内容副本可以缓存在距离请求用户更近的中间节点上,为未来潜在的用户请求提供就近响应的机会,从而大大减小了内容获取的延迟和带宽需求,并减轻了内容原始服务器的访问压力.因此,针对内置缓存机制的研究一直是 CCN 领域关注的热点之一.

近年来,研究人员针对 CCN 环境下的内容缓存机制开展了大量的研究.其中,最简单的当属作为 CCN 默认缓存策略的 LCE(Leave Copy Everywhere)算法^[3].然而,该算法采用了处处缓存的方式,将内容向其所经过的每一个中间节点进行复制,从而导致了大量的缓存替换和内容冗余.为了改进缓存的冗余性等问题,继 LCE 之后一系列的优化方案被陆续提出,其大致可分为集中式策略和分布式策略两大类.文献^[4]提出了一种集中式的内容缓存策略 OFAM,通过一个被称为 CM(Cache Management)的管理节点来实现域内的缓存决策.与许多分布式的内容缓存策略通过内容交付路径上的节点自主地进行内容缓存决策不同,该机制中,统一由 CM 根据对用户请求的统计估算各个内容分组的流行度,然后选择流行度高的内

容进行缓存.其他典型的集中式内容缓存策略还包括 ICAP^[5]、C-flow^[6]等等.在集中式缓存策略中,缓存的决策以及内容副本的存储往往都过分依赖于少量的特殊节点,导致这部分节点的负载过大,而其他节点的资源则得不到有效利用,难以满足实际网络的性能需求.因此,分布式缓存策略成为了目前 CCN 缓存机制研究的主要方向.

根据缓存决策的依据不同,现有的 CCN 分布式缓存策略主要可划分为以下 3 类.第 1 类缓存机制采用随机方式选择内容的缓存节点,如文献^[7]提出了一种分布式的缓存随机放置策略,以一定的概率在内容对象返回路径上的节点中随机地选择缓存节点.随机决策策略在一定程度上降低了内容缓存的冗余度,但是却无法保证缓存节点选择的最优性.第 2 类缓存机制从内容的相关特性出发实现缓存的合理决策.如,文献^[8]提出了根据内容的流行度来进行缓存决策的 WAVE 缓存机制,为高流行度的内容提供更多的缓存机会.文献^[9]针对多级多域的层次型网络结构提出了一种缓存放置策略.该策略中,节点为每个内容设置一个独立的计数器 RC,每收到一个内容的外生请求或域间请求,则相应内容的 RC 计数值加 1,当内容的 RC 计数值超过某个阈值时,则节点选择该内容进行缓存;另一方面,每个内容的 RC 计数器每隔一定的时间间隔会自动减 1,当计数值降为 0 时,则节点丢弃相应的内容.与之类似,文献^[10]提出了一种基于优先级的概率性缓存机制 PPC,该机制中,内容被划分为多个数据分组,节点根据每个数据分组对于重构内容的重要性评估其相应的优先级,然后结合流行度来计算各数据分组的缓存概率,确保流行度和重要性较高的数据分组获得更大的缓存概率.该类型其他的典型机制还包括 DCPCM(Dynamical Content Popularity for Cache Management)^[11],DCDS(Diverse Content Delivery Scheme)^[12],CDBPC(Content Diffusion Based Proactive Caching)^[13]等.然而,这类机制往往忽略了诸如节点重要性、缓存状态等重要信息,因此,大大影响了算法的有效性.第 3 类缓存机制则考虑根据节点的相关属性来选择最佳的缓存节点. Chai 等人^[14]提出利用节点重要性选择内容的缓存位置,设计了基于节点中心性的选择性缓存算法 Betw,选择内容交付路径上介数值最大的节点作为缓存节点.然而该机制未考虑节点的可用缓存容量以及负载情况等网络动态信息,而节点介数则是一个相对稳定的

属性.因此,缓存内容大量集中在高介数节点造成其余节点缓存资源的严重浪费.部分研究人员试图结合节点和内容的特性来改进 CCN 缓存决策机制的性能.如,文献[15]提出了一种基于内容流行度和节点等级的概率性缓存机制 MPC,通过综合考虑节点与内容请求节点跳数距离、节点的介数中心性等信息来划分节点的等级,然后在缓存决策时,节点将以更大的概率缓存流行度与其节点等级相匹配的内容,从而在实现缓存利用率最大化的同时提高缓存内容的多样性.Li 等人^[16]则提出了一种轻量级的在线缓存机制 RBC-CC.该机制在考虑内容年龄的基础上提出了加权内容流行度的概念,以反映内容流行度随时间的变化.进而,综合利用内容的流行度和路由介数中心性来实现内容的缓存放置,通过将流行内容尽可能地缓存在介数中心性较高的重要节点来达到提高缓存命中率、降低内容访问延迟的目标.然而,由于在缓存决策时缺乏对节点缓存能力的考虑,容易导致缓存内容因缓存空间限制在响应用户请求之前即被过早替换的无效缓存问题.Psaras 等人^[17]提出了一种概率性缓存机制 ProbCache,综合考虑内容交付路径上的累积缓存容量以及与原始内容服务器间的距离设计了一个效用函数来实现缓存节点的决策.Lee 等人^[18]根据内容响应路径上节点的可用缓存容量,选择可用缓存资源最多的节点进行内容缓存,从而在降低缓存冗余的同时实现节点缓存的均匀利用.这类算法虽然考虑了节点的缓存能力,但是其问题是在网络负载较大而节点缓存资源有限的情况下,多数节点的缓存都将逐渐接近并达到饱和.因此,单一地利用可用缓存空间并不一定能准确地反映节点间的缓存状态差异.

针对现有 CCN 缓存机制存在的上述问题,本文综合了节点的缓存占用率、缓存替换率、内容的流行度以及剩余订阅时间等信息来评估缓存节点选择的合理性和内容的缓存价值,提出了一种基于缓存压力的选择性缓存机制.其主要贡献主要体现在以下方面:

1)设计了一种基于节点缓存压力的缓存节点选择算法,利用节点的缓存占用率和缓存替换率评估节点的缓存压力,并结合内容的流行度实现缓存节点的合理选择;

2)提出了一种缓存替换算法,利用内容的流行度以及剩余订阅时间评估内容的缓存价值,从而对缓存价值较低的内容实现优先替换.

1 CPBSC 缓存机制

CPBSC 机制主要包括 2 个部分:缓存节点选择

和缓存内容替换.在缓存节点选择算法中,我们首先估算各节点的缓存压力(Cache Pressure, CP),内容兴趣分组在传播过程中会记录其转发路径上经过节点的最小缓存压力值,然后在响应数据分组返回时,再根据节点缓存压力和内容的流行度合理地选择缓存节点.而在缓存节点空间不足时,则根据内容的流行度和剩余订阅时间对内容的缓存价值进行排序,然后优先丢弃缓存价值低的内容.

1.1 缓存压力的估计

节点的缓存状态是缓存节点选择的重要依据,可用缓存空间或缓存占用率是反映节点缓存状态的常见测度之一.然而,在重负载的情况下,节点的缓存大多处于或接近饱和,即可用缓存空间为零(缓存占用率为 100%).此时,单一地通过可用缓存空间或缓存占用率无法准确地区分各个节点的实际缓存状态.考虑到节点缓存饱和时,新到达的内容数据分组可能触发缓存内容的替换操作,因此,缓存替换率可以作为节点缓存状态的补充测度,弥补可用缓存空间(或缓存占用率)在网络重负载下的不足.

我们首先定义节点 v 的缓存占用率 $\text{Occ}(v)$ 为节点当前缓存的内容大小与节点缓存总容量 $C(v)$ 的比值,即

$$\text{Occ}(v) = \frac{\sum_{i=1}^n S(c_i)}{C(v)} \quad (1)$$

式中: $S(c_i)$ 为节点当前缓存的内容 c_i 的大小; n 为节点当前缓存的内容个数.

缓存替换率则定义为单位采样时间内节点缓存中被替换的内容大小与节点缓存总容量的比值,即节点 v 的缓存替换率 $\text{Rep}(v)$ 可表示为:

$$\text{Rep}(v) = \frac{\sum_{j=1}^m D(c_j)}{C(v)} \quad (2)$$

式中: $D(c_j)$ 为节点 v 中被替换的内容 c_j 的大小; m 为单位采样时间内节点 v 缓存中替换的内容个数.

然后,利用节点的缓存占用率结合缓存替换率来计算节点 v 的缓存压力值 $\text{CP}(v)$.其具体的计算方法如式(3)所示.

$$\text{CP}(v) = \text{Occ}(v) + \text{Rep}(v) \quad (3)$$

这样,在网络负载较轻(节点缓存未饱和)时,替换率 $\text{Rep}(v)$ 可能为 0,节点的缓存压力主要由其缓存占用率决定;而在网络负载较重(节点缓存已饱和)时,节点缓存占用率 $\text{Occ}(v)$ 即便均接近或等于 1,但节点缓存压力仍可通过缓存替换率 $\text{Rep}(v)$ 体现不同节点在缓存状态上的差异.

我们在内容的兴趣分组头部增加 1 个节点缓存

压力字段 CPV, 用来记录节点缓存压力的估计值, 该字段初始值设为一个较大的值, 如 1 000. 兴趣分组转发过程中, 每经过一个中间路由节点, 均会将 CPV 的值与当前节点的缓存压力值比较并更新为较小值. 当发生缓存命中时, 内容响应节点将兴趣分组 CPV 字段记录的最小缓存压力值 CP_{\min} 写入到对应的数据分组中并返回.

1.2 选择性缓存

当响应数据分组沿着其兴趣分组的反向路径返回请求用户节点时, 将依据各中间节点的缓存压力值和内容的流行度选择合适的缓存节点. 这里, 一方面尽可能将内容副本缓存在缓存压力较小的节点上, 以保证节点资源的合理均匀利用; 另一方面, 现有的研究表明, 用户对内容的请求具有一定的偏好性, 且用户请求服从 Zipf 分布. 网络中较流行(请求用户较多)的内容由于受关注程度较高, 应该在缓存决策时得到更多的缓存机会来满足其较大的潜在需求.

然而, 内容的流行度具有全局相关性, 它与网络中所有用户节点的内容请求状态有关. 但在实际网络环境中, 全局信息的获取往往需要付出巨大的额外开销. 为此, 我们首先利用内容在节点本地的请求到达率来近似估计相应内容的流行度. 这里, 将节点 v 处内容 c_i 的请求到达率定义为单位采样时间内节点 v 收到的关于内容 c_i 的兴趣分组数量, 记为 $\lambda_v(c_i)$. 然后, 将节点 v 处内容 c_i 的请求到达率与节点 v 收到的总请求到达率的比值定义为内容 c_i 在节点 v 处的流行度权重因子 $\theta_v(c_i)$, 即

$$\theta_v(c_i) = \frac{\lambda_v(c_i)}{\sum_{k=1}^n \lambda_v(c_k)} \quad (4)$$

当内容 c_i 的响应数据分组到达中间路由节点 v 时, 节点 v 根据响应数据分组中携带的最小缓存压力值 CP_{\min} 以及本地计算得到的关于内容 c_i 的流行度权重因子 $\theta_v(c_i)$, 计算节点本地的缓存压力阈值 $CP_{th}(v)$, 即

$$CP_{th}(v) = CP_{\min}(1 + \theta_v(c_i)) \quad (5)$$

节点将当前的缓存压力值与该阈值比较, 如果节点当前缓存压力小于该阈值则将当前节点选择为内容的缓存节点. 否则, 节点直接向下一跳节点转发而不做缓存处理.

1.3 缓存替换

当某个节点被选定为内容的缓存节点时, 内容副本将向缓存节点进行复制. 然而, 若缓存节点当前空间不足, 则需进行缓存替换决策. 为了保证节点缓存资源的有效利用, 对内容的缓存价值进行评估, 优

先替换缓存价值较低的内容而尽可能地延长高缓存价值内容的缓存时间. 如前所述, 用户未来对高流行度内容的潜在需求较大, 缓存高流行度的内容能更好地为用户后续请求提供响应服务. 因此, 内容的缓存价值一方面应该与内容的流行度成正比. 这里, 内容的流行度仍然通过节点本地的请求到达率来计算. 然而, 用户对内容的请求也存在一定的时间限制, 体现在每个兴趣分组均存在一个生存时间 (Time To Live, TTL), 当 TTL 到期时, 该兴趣分组将被直接丢弃, 相应用户对内容的请求也随之失效. 由于不同用户节点生成兴趣分组的时间可能不同, 因而来自不同订阅节点的对同一内容的兴趣分组也可能存在完全不同的剩余生存时间 (Residual Time To Live, RTTL). 考虑到兴趣分组的 RTTL 越短, 其发生失效的可能更大, 通过内容缓存响应用户请求服务的机会也越低, 因此, 我们将不同用户节点对某一内容的兴趣分组的平均剩余生存时间作为该内容缓存价值的另一评价依据. 即内容兴趣分组的平均剩余生存时间越长, 其缓存价值越大, 反之亦然.

由此, 我们将任意内容 c_i 对于节点 v 的缓存价值 $CV_v(c_i)$ 定义为节点 v 处内容 c_i 的请求到达率 $\lambda_v(c_i)$ 与节点 v 收到的内容 c_i 的兴趣分组的平均剩余生存时间的乘积. 即

$$CV_v(c_i) = \lambda_v(c_i) \frac{\sum RTTL_j}{L} \quad (6)$$

式中: L 为节点 v 收到的来自不同用户对内容 c_i 的兴趣分组数量; $RTTL_j$ 为第 j 个用户节点的兴趣分组的剩余生存时间.

节点在缓存替换决策时, 首先对各个内容的缓存价值进行计算, 然后优先替换掉其中缓存价值最低的内容副本.

2 性能评估

为了证明 CPBSC 缓存机制在缓存性能上的优势, 我们选择了处处缓存的 LCE 策略、基于介数的 Betw 策略以及基于缓存容量的概率性缓存策略 ProbCache 作为 CPBSC 性能比较的对象, 并利用 ndnSIM^[19] 模拟器实现了以上 4 种策略的性能仿真. 仿真实验针对在缓存大小、内容数量和 Zipf 参数 α 等网络参数变化的情况下, 各缓存策略主要性能指标的差异进行了定量分析和比较. 本文采用的主要性能评估指标包括:

1) 缓存命中率. 缓存命中率是评价缓存性能的重要指标之一, 它被定义为用户请求由缓存而非原始内容服务器响应的概率. 缓存内容的命中率越高,

对应的原始内容服务器响应率和负载就越小,系统的效率就越高。

2)平均接入代价.平均接入代价被定义为用户请求找到匹配的内容对象所需的平均跳数,它反映了用户请求得到响应,用户完成内容获取的速度.平均接入代价越小,内容获取速度越快,缓存系统效率越高。

2.1 实验参数设置

本文利用 GT-ITM 生成了一个由 50 个路由节点组成的仿真网络拓扑.假设用户请求的到达过程服从泊松分布,用户的访问模式服从参数为 α 的 Zipf 分布.各节点缓存大小相同且初始为空.兴趣分组采用洪泛方式进行转发,除 CPBSC 外,其他 3 种缓存机制均采用 LRU 作为默认的缓存替换策略.其他主要参数如表 1 所示,在无特殊说明时,各实验参数均取默认值。

表 1 实验参数设置
Tab. 1 Parameter settings

主要参数	默认值	变动范围
内容数量/个	2 000	100~5 000
内容大小/MB	1	
请求到达速率/(个·s ⁻¹)	50	
节点缓存大小/MB	100	5~1 000
用户访问模式	Zipf, $\alpha=0.8$	0.6~1.2

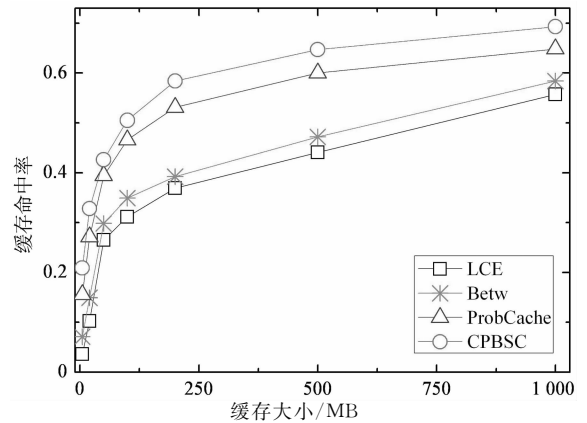
2.2 仿真实验结果

为了清晰地反映上述 4 种不同缓存机制的性能差异,在仿真实验中,通过分别改变节点的缓存大小和内容数量来分析缓存资源对各项性能指标的影响.然后,考虑到不同应用下用户访问模式的 Zipf 分布参数 α 是存在差异的,因此,我们调整 α 的大小来研究不同缓存机制面向不同应用时的表现.具体实验结果如下。

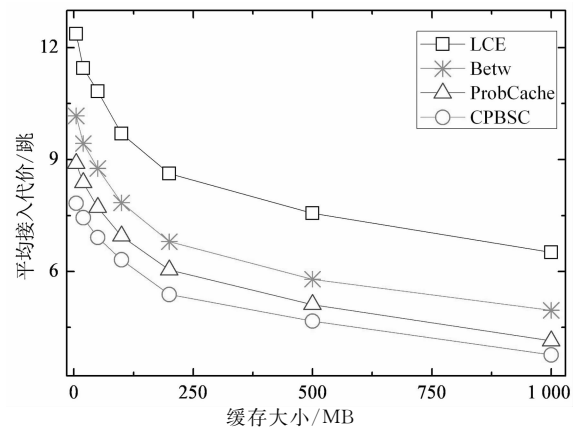
2.2.1 缓存大小的影响

本小节我们首先来研究采用不同缓存策略时系统缓存命中率随网络参数变化的情况,仿真结果如图 1 所示.由图 1(a)可以看到,随着节点缓存容量的增加,4 种机制的缓存命中率都呈现出增长的趋势.这是由于节点缓存空间增加使得内容分组在网络中的缓存时间延长,从而提高了缓存命中的概率.其中,LCE 较为盲目激进的缓存策略造成严重的内容冗余和频繁的内容更替,大大降低了系统的缓存性能,因此在图中缓存命中率最低;Betw 利用节点的介数优化了缓存节点的选择,但缓存内容过度集中于高介数节点,导致缓存替换频繁,影响了缓存命中性能,因此缓存命中率仅略高于 LCE;ProbCache 通过对内容交付路径上的累积缓存空间的估计来进行概率性的缓存节点选择,一定程度上实现了对缓

存资源的均匀利用,从而其缓存命中率高于 LCE 和 Betw;而 CPBSC 获得了四者中最优的缓存命中性能,在节点缓存为 100 MB 时,其缓存命中率达到 50.5%,比 ProbCache、Betw 和 LCE 分别高出约 8.4%、44.7%和 62.3%。



(a) 缓存命中率与缓存大小之间的关系



(b) 平均接入代价与缓存大小之间的关系

图 1 节点缓存对缓存性能的影响

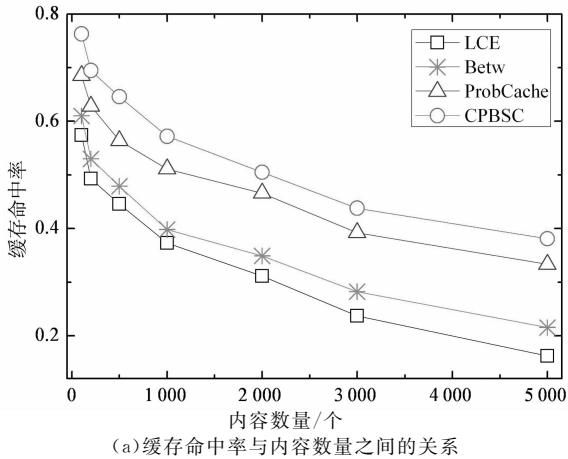
Fig. 1 Impact of cache size on caching performance

同样,随着节点缓存的增加,中间节点对内容的缓存能力增强,每个内容分组能获得更长时间的缓存服务.这意味着用户有更大的可能从距离较近的中间缓存节点实现快速的内容获取.因此,图 1(b)中各机制的平均接入代价均随着节点缓存的增加而逐渐降低.其中,得益于对节点缓存压力的准确估计以及内容流行度的综合考虑,CPBSC 对缓存内容的放置和替换更加合理,用户可以更多地通过内容缓存实现快速的内容获取,因此,CPBSC 的平均接入代价最小.以节点缓存为 100 MB 时的情况为例,ProbCache、Betw 和 LCE 的平均接入代价达到了 6.95 跳、7.84 跳和 9.69 跳,而 CPBSC 仅为 6.31 跳,与前三者相比,分别减少了近 9.2%、19.5%和 34.9%。

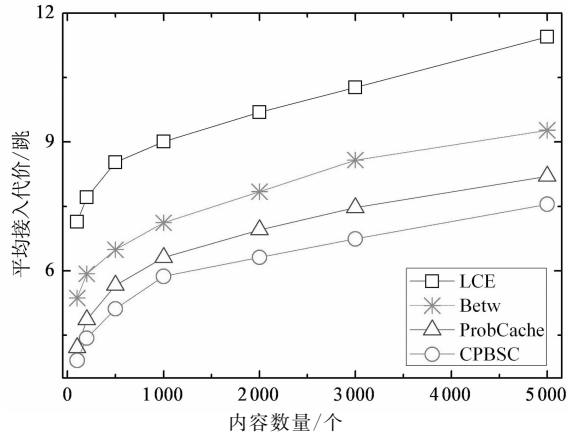
2.2.2 内容数量的影响

接下来,我们就内容数量对 4 种缓存机制的影响进行分析,试验结果如图 2 所示.由于在节点缓存

大小固定的情况下,内容数量的增加意味着可用缓存资源的相对紧张,因此,内容数量对缓存性能的影响实际上也是节点缓存大小与缓存性能之间关系的另一种体现。



(a)缓存命中率与内容数量之间的关系



(b)平均接入代价与内容数量之间的关系

图 2 内容数量对缓存性能的影响

Fig. 2 Impact of content number on caching performance

由图 2(a)可知,随着内容数量的增多,节点缓存资源越来越紧缺,4 种机制的缓存命中率均出现了明显下降的趋势。然而,CPBSC 始终保持了四者中最高缓存命中率,由于 LCE 机制下严重的冗余缓存,使其即使在缓存相对充足(内容数量为 100 个)时也仅获得了 57.4% 的缓存命中率,而 CPBSC 则达到了 76.3%。当内容数量增加到 5 000 个时,CPBSC 仍能获得 38.1% 的缓存命中,相比 ProbCache(33.3%)、Betw(21.5%)和 LCE(16.2%)分别提高了约 14.4%、77.2%和 135.2%。

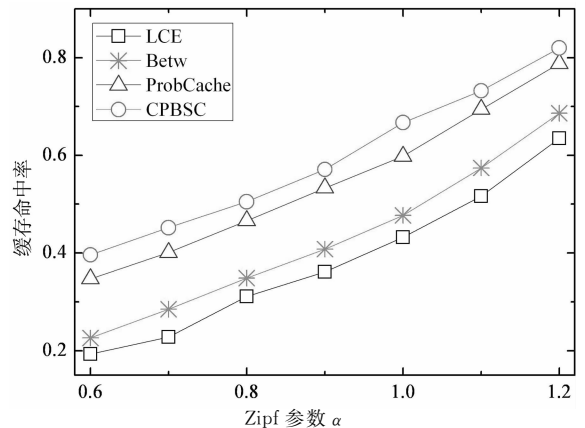
由图 2(b)可知,4 种缓存机制的平均接入代价则随着内容数量的增加逐渐增大,这是由于内容数量的增加意味着内容缓存需求的增大,在节点缓存资源有限的情况下,缓存替换增多使得无效缓存越来越严重,导致通过内容缓存加快用户响应的作用逐渐减弱。尽管如此,CPBSC 的平均接入代价明显低于其他 3 种缓存机制。同样以 5 000 个内容时的

情况为例,CPBSC 的平均接入代价仅为 7.55 跳,相比 ProbCache、Betw 和 LCE 分别减少了约 7.9%、18.6%和 34.1%。这与上节中关于缓存大小对两种缓存性能指标的影响的分析结果是一致的。

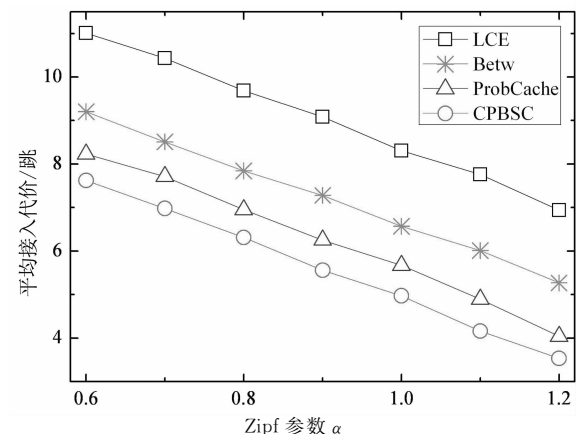
2.2.3 用户访问模式的影响

用户对内容的访问具有一定的偏好性,本小节我们进一步研究用户偏好对不同机制的缓存命中率的影响,如图 3 所示。Zipf 参数 α 越大,意味着用户的偏好越发地向流行度高的内容集中。由于 3 种缓存机制均采用了优先保证高流行度内容缓存时间的相关策略,因此,由图 3(a)可见,它们的缓存命中率随着 α 值的增大均呈现上升的趋势。其中,CPBSC 在 α 取不同值时的缓存命中率相比其他 3 种缓存机制均具有明显的优势。在 α 取值为 0.6 时,CPBSC 的缓存命中率相比 LCE、Betw 和 ProbCache 分别提高了约 105.2%、75.2%和 14.1%,而在 α 值增大到 1.2 时,其缓存命中率的提高比例也分别达到了近 63.5%、19.5%和 4.1%。

在平均接入代价方面,从图 3(b)中的结果可以发现用户偏好性的增强导致了平均接入代价的下



(a)缓存命中率与 Zipf 参数 α 之间的关系



(b)平均接入代价与 Zipf 参数 α 之间的关系

图 3 Zipf 参数 α 对缓存性能的影响

Fig. 3 Impact of Zipf parameter α on caching performance

降.这是由于流行度高的内容得到了较多的缓存机会,随着 α 值的增大,用户请求对高流行度内容的偏向性使得这部分请求更多地通过中间缓存节点得到满足,因此平均接入代价逐渐减小,响应速度提高.同样,CPBSC的平均接入代价明显低于其他缓存机制,获得了最快的内容获取速度.

3 结 论

为了合理、充分地利用网络的缓存资源,提升系统的缓存性能,本文提出了一种基于缓存压力的选择性缓存机制 CPBSC.该机制利用节点的缓存占用率及替换率较准确地估计节点的缓存压力,并结合内容流行度实现了缓存内容的合理放置.同时,根据内容的请求率和兴趣分组的剩余生存时间评估不同内容的缓存价值,以此实现缓存内容的合理替换.仿真结果表明 CPBSC 有效地优化了缓存资源利用率,并显著提高了系统的整体缓存性能.在今后的工作中,我们将对 CPBSC 在实际网络环境下的性能进行验证和优化,同时进一步研究如何将其扩展到移动网络以及其他复杂的网络环境.

参考文献

- [1] JACOBSON V, SMETTERS D K, THORNTON J D, *et al.* Networking named content[J]. *Communications of the ACM*, 2012, 55(1): 117–124.
- [2] AHLGREN B, DANNEWITZ C, IMBRENDA C, *et al.* A survey of information-centric networking[J]. *IEEE Communications Magazine*, 2012, 50(7): 26–36.
- [3] LAOUTARIS N, SYNTILA S, STAVRAKAKIS I. Meta algorithms for hierarchical web caches[C]//*Proceedings of the IEEE International Performance Computing and Communications Conference (IEEE IPCCC)*. Piscataway, NJ: IEEE, 2004: 445–452.
- [4] JMAL R, FOURATI L C. An open flow architecture for managing content-centric-network (OFAM-CCN) based on popularity caching strategy[J]. *Computer Standards & Interfaces*, 2017, 51: 22–29.
- [5] LIU T, ABOUZEID A A. Content placement and service scheduling in femtocell caching networks[C]//*Proceedings of the 59th IEEE Global Communications Conference (GLOBECOM)*. Piscataway, NJ: IEEE, 2016: 1–6.
- [6] CHANG D, KWAK M, CHOI N, *et al.* C-flow: An efficient content delivery framework with OpenFlow[C]// *Proceedings of the 28th IEEE International Conference on Information Networking (ICOIN2014)*. Washington DC: IEEE Computer Society, 2014: 270–275.
- [7] 胡骞,武穆清,郭嵩,等.一种用于内容中心网络的缓存随机放置策略[J]. *西安电子科技大学学报(自然科学版)*, 2014, 41(6): 131–136, 187.
- [8] HU Q, WU M Q, GUO S, *et al.* Random cache placement strategy for content-centric networking[J]. *Journal of Xidian University(Natural Sciences)*, 2014, 41(6): 131–136, 187. (In Chinese)
- [9] CHO K, LEE M, PARK K, *et al.* WAVE: Popularity-based and collaborative in-network caching for content-oriented networks[C]//*Proceedings of the IEEE INFOCOM 2012 Computer Communications Workshops*. Piscataway, NJ: IEEE, 2012: 316–321.
- [10] DOMINGUES G, EDSE S, RMM L, *et al.* Enabling opportunistic search and placement in cache networks[J]. *Computer Networks*, 2017, 119: 17–34.
- [11] SIRICHOTEDUMRONG W, KUMWILASAK W, TARNOI S, *et al.* Prioritized probabilistic caching algorithm in content centric networks[C]//*Proceedings of the 12th International Conference on Computing and Information Technology*. Berlin: Springer-Verlag, 2016: 255–265.
- [12] 葛国栋,郭云飞,刘彩霞,等. CCN 中基于业务类型的多样化内容分发机制[J]. *电子学报*, 2016, 44(5): 1124–1131.
- [13] GE G D, GUO Y F, LIU C X, *et al.* A diverse content delivery scheme based on traffic types in content centric networking [J]. *Acta Electronica Sinica*, 2016, 44(5): 1124–1131. (In Chinese)
- [14] 张果,汪斌强,张震,等. 基于节点动态内容流行度的缓存内容管理策略[J]. *电子学报*, 2016, 44(11): 2703–2711.
- [15] ZHANG G, WANG B Q, ZHANG Z, *et al.* A strategy based on dynamical content popularity for cache management[J]. *Acta Electronica Sinica*, 2016, 44(11): 2703–2711. (In Chinese)
- [16] 罗熹,安莹,牛碧诺. 内容中心网络中一种基于内容扩散的主动缓存机制[J]. *湖南大学学报(自然科学版)*, 2016, 43(8): 142–150.
- [17] LUO X, AN Y, NIU B N. Content diffusion based proactive caching in content-centric networks[J]. *Journal of Hunan University(Natural Sciences)*, 2016, 43(8): 142–150. (In Chinese)
- [18] CHAI W K, HE D, PSARAS I, *et al.* Cache “less for more” in information-centric networks (extended version)[J]. *Computer Communications*, 2013, 36(7): 758–770.
- [19] LI Y, ZHANG T, XU X, *et al.* Content popularity and node level matched based probability caching for content centric networks[C]//*Proceedings of IEEE/CIC International Conference on Communications in China*. Piscataway, NJ: IEEE, 2016: 1–6.
- [20] LI J, WU H, LIU B, *et al.* RBC-CC: RBC-based cascade caching scheme for content-centric networking[J]. *Journal of Network & Systems Management*, 2017, 25(2): 375–396.
- [21] PSARAS I, CHAI W K, PAVLOU G. Probabilistic in-network caching for information-centric networks[C]//*Proceedings of the 2nd ICN Workshop on Information-Centric Networking*. New York: ACM, 2012: 55–60.
- [22] LEE J, LIM K, YOO C. Cache replacement strategies for scalable video streaming in CCN[C]// *Proceedings of the 19th Asia-Pacific Conference on Communications (APCC 2013)*. Washington DC: IEEE Computer Society, 2013: 160–165.
- [23] MASTORAKIS S, AFANASYEV A, MOISEENKO I, *et al.* ndnSIM 2.0: A new version of the NDN simulator for NS-3 [R]. Los Angeles: University of California, 2015. <http://named-data.net/techreport/ndn-0028-1-ndnsim-v2.pdf>.