

基于边缘优先的 ICN 缓存协作策略

智江^{1,2}, 李俊¹, 吴海博¹, 任勇毛¹

(1. 中国科学院计算机网络信息中心, 北京 100190; 2. 中国科学院大学, 北京 100049)

摘要: 针对信息中心网络 (ICN) 环境下缓存内容放置优化问题, 提出一种边缘优先逐级反馈的缓存协作策略 (ECCS)。该策略通过将缓存决策提前至请求转发阶段, 将下游节点的缓存决策信息及内容统计信息逐级反馈给上游节点, 以达到缓存信息即时更新、辅助上游节点完成协作缓存的目的。实验结果表明, ECCS 在服务负载减少率、平均跳数减少率和平均缓存命中率等方面与现有方法相比有显著优势。

关键词: 信息中心网络; 网内缓存; 内容放置; 边缘优先; 协作缓存

中图分类号: TP393

文献标识码: A

Edge-first-based cooperative caching strategy in information centric networking

ZHI Jiang^{1,2}, LI Jun¹, WU Hai-bo¹, REN Yong-mao¹

(1. Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China;

2. University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: In order to optimize the replica placement in information centric networking, an edge-first-based cooperative caching strategy (ECCS) was proposed. According to the strategy, cache decision was made during the interest forwarding stage. The decision result and statistic information would be forwarded to upstream routers step by step. Utilizing the information, upstream nodes could update their cache information table immediately to achieve cooperative caching. The experimental results indicate ECCS can achieve salient performance gain in terms of server load reduction ratio, average hop reduction ratio, average cache hit ratio, compared with current strategies.

Key words: information centric networking, in-network caching, content placement, edge priority, cooperative caching

1 引言

随着互联网的高速发展, 大量网络的应用随之产生, 其中, 内容获取类应用占据了重要地位。根据 Cisco VNI 报告的预测, 2018 年视频类应用产生的网络流量将占网络总流量的 79%。而 TCP/IP 面向连接的设计思想却与当前的主要应用模式产生了矛盾。为了解决此问题, 研究者们提出了以信息为中心的全新的网络架构——信息中心网络 (ICN, information centric networking)^[1], 其中以 CCN/NDN^[2]、DONA^[3]、

NetInf^[4]、PURSUIT^[5]等为典型代表。与传统 TCP/IP 架构不同, 这种全新的网络架构将内容作为网络的第一元素, 网络寻址的目标不再是主机而是内容, 终端网络节点通过发送包含内容名字的分组直接获取内容, 而不必将内容名转换为具体主机地址, 具体的寻址及路由的工作交给网络层来做, 从而解放应用层。这种全新的网络架构带来一个明显的好处就是可以方便地通过缓存来改善网络性能。由于网络中的分组由原先标识地址的 IP 数据分组转变成了标识内容名字的分组, 因此, 对于缓存系统来说无需再做高层协

收稿日期: 2016-08-08; 修回日期: 2017-01-06

通信作者: 李俊, lijun@cnic.cn

基金项目: 国家自然科学基金资助项目 (No.61672490, No.61602436, No.61601443); 北京市自然科学基金资助项目 (No.4174110); 中国科学院重大装备研制基金资助项目

Foundation Items: The National Natural Science Foundation of China (No.61672490, No.61602436, No.61601443), The Natural Science Foundation of Beijing (No.4174110), The Significant Equipment Development Program of Chinese Academy of Sciences

议分析等工作,而可以直接对数据分组进行缓存,通过名字来查询和匹配一个具体的数据,这使缓存系统的构建变得更加方便快捷。另外,由于内容获取类应用的特点,用户请求具有时间局部性和空间局部性特征,因此,将缓存系统加入网络架构有助于提升网络性能,改善用户体验,二者相得益彰,网络缓存在未来必然有更为广泛的应用。

在实际部署中,网络缓存设备放置之后,位置、容量等属性不可能频繁变化。而用户的访问行为以及内容的更新却有很强的动态性。因此,如何通过已有的缓存设备中放置内容来优化缓存性能成为一个实际存在的问题。以 ICN 中的代表架构命名数据网络(NDN, named data networking)为例介绍网内缓存的特点及存在的问题。

对于网络缓存决策来说,要解决的问题可以归结为:1) 选择哪些内容;2) 放置到哪些节点上。

为了满足线速转发的要求,NDN 路由器中缓存容量非常有限,只有选择热度较高的内容才能发挥出缓存的优势。另外,在多数实际场景中,位于网络核心及中间的路由器不产生请求,外部请求来自边缘路由器。当边缘节点缓存某内容之后,其转发的请求随之消失,此节点上游的节点收到对该内容的请求的几率会随之降低,上游路由器缓存该块内容带来的收益也会下降。因此,缓存位置的选择也是决定缓存效果的重要影响因素。

本文利用了 NDN 中内容的请求及返回遵循相同路径的特点,提出一种边缘优先逐级反馈的缓存协作策略,通过路径上的协作决定缓存的内容和缓存的位置,提高缓存系统的整体性能。

本文的主要贡献如下。

1) 对内容放置问题进行建模研究,对内容热度与内容位置进行综合分析,提出边缘优先放置的思想。

2) 提出边缘优先逐级反馈的轻量级缓存协作机制(ECCS)。

3) 定义了适合 ICN 特性的性能评价指标,通过实验验证了 ECCS 策略在缓存性能上的优势。

2 相关工作

对缓存内容的放置问题可以抽象为在哪里存、存哪些内容的问题,以下从这 2 个角度对目前的研究工作做出一些简要的介绍和分析。

2.1 缓存位置决策

ICN 默认采用随处缓存(CEE, cache everything everywhere)^[2]的方案,内容在返回途中被沿途所有缓存节点缓存。该方案没有对缓存位置和缓存内容做出任何选择,会缓存大量无用的冷门内容,同时,会产生内容冗余,难以充分发挥缓存的效果。

Laoutaris 等^[6]提出一种 LCD(leave copy down)缓存方案,以提高缓存空间利用率。当请求命中缓存时,仅在命中节点的直接下游节点缓存此内容,用以降低缓存内容的冗余度。文献[6]同时提出另一种 MCD(move copy down)缓存方案,与 LCD 的区别在于缓存命中时除向下游节点缓存内容外,同时删除当前节点的缓存内容。上述 2 种方法均是针对 Web 缓存提出的解决缓存冗余度的早期方法。以上 2 种方案旨在通过限制同样内容沿路复制数量来减少内容冗余,然而通过实验发现在某些复杂网络拓扑下,这 2 种方案并不能很好地降低内容冗余度,反而会使缓存性能降低。

Chai 等^[7]提出一种基于介数(betweenness)的选择性缓存机制,通过沿路计算节点的介数,选择介数最大的节点作为内容缓存的节点,从而提高缓存命中率,减少替换次数和内容冗余度。

Psaras 等^[8]设计了一种基于概率的缓存机制(probcache),通过综合考虑缓存容量以及缓存收益,确定每一个节点的缓存概率,并以此概率缓存内容。其概率的选择正比于缓存节点的容量,反比于缓存节点距请求者的距离。

葛国栋等^[9]提出一种 CCN 中基于替换率的缓存空间动态借调机制,依据节点对存储资源的需求程度对缓存资源进行动态借调,将部分内容转存在相对空闲的存储资源中来提升整体缓存性能。

以上方案从不同角度入手,主要解决在哪里存的问题,然而它们对所存的内容未加区分,因此,可能造成大量冷门的内容占据缓存,无法充分发挥缓存的效果。

2.2 缓存内容决策

Wang 等^[10]提出一种同时考虑内容访问频率和缓存距离收益的替换算法 LB(least benefit),每次替换收益最小的内容。具体而言,在每次缓存命中时,将本次收益值计入总收益中;当发生内容替换时,每次替换收益值最小的内容。该方法同时考虑单次缓存收益以及访问频率。该方法可以获取优于 LFU(least frequently used)算法的性能,然而两者相

差不多；同时，由于需要维护收益信息，会引入一定的额外开销。

刘外喜等^[11]提出一种选择性缓存机制(SC, selective caching)，根据内容传播的规律将内容分为传播早期和传播晚期，通过记录内容在节点被请求的历史，推断内容处于传播的哪个时期，根据内容所处的不同时期计算相应的缓存概率，同时融入了以带宽换缓存的思想。

以上几种策略考虑了缓存内容的热度等收益信息，但缺乏对缓存位置信息的考虑，因此难以发挥全部的缓存性能。

2.3 缓存位置决策&缓存内容决策

Ming 等^[12]提出了一种基于内容生存期(age)的方案，只有当内容生存期超时才有可能被替换。内容生存期由存储位置和内容热度共同决定，即内容越靠近网络边缘，内容热度越高，生存期越长。内容分组携带生存期字段，沿途各路由节点借助和调整此数值来决定内容生存期。该方法兼顾了内容放置和内容替换，然而，有效期需要精确设计，否则会影响内容替换和性能。

Cho 等^[13]提出一种 WAVE 缓存机制，上游节点利用块标记窗口给下游节点提供缓存建议，块标记窗口会随着内容的流行程度而呈指数增长，从而使越流行的内容分布越广，热门内容快速扩散，同时又能有效阻止冷门内容的分发。然而，该方案在记录文件访问次数时，以及在上下游节点间传输缓存建议时，均会引入额外开销，并且缓存建议开销会随着文件块数的增长而增大。

Ren 等^[14]提出一种最大化收益的缓存策略(MAGIC)通过在请求路径上计算缓存内容所带来的收益来决定具体的缓存位置。

Wu 等^[15]提出一种基于最大化缓存收益的概率缓存策略(MBP)，综合考虑内容热度和缓存放置收益计算求得内容在节点上的缓存概率，以此概率缓存该内容。请求热度和缓存收益高的内容将有更高的概率缓存于此节点，以提升整体缓存性能。

以上策略从不同角度综合考虑了缓存位置和缓存内容，但是它们都未考虑缓存节点间决策的相互影响，缺乏整体的视角，因此缓存性能仍有提升的空间。

3 问题描述与分析

NDN 是 ICN 体系结构中的典型代表，因此以 NDN 为例对本文所提出的策略进行阐释。相关符号

说明如表 1 所示。

NDN 中默认路由机制采用 SPT 类路由算法生成路由表，如 OSPFN^[16]或 NLSR^[17]。路由器通过查询路由表将用户请求向内容源的方向转发，如果路径中遇到网内缓存中存在请求的内容，则由缓存命中节点直接响应。内容数据分组根据 PIT 表中记录的信息按照原路返回给请求者。每个 NDN 路由节点包含一定容量的网内缓存并运行相应的缓存策略。缓存的对象以数据块(chunk)为单位，这里，假设每个 chunk 大小相同。

表 1 符号说明

符号	含义
V	缓存节点集合
F	内容块集合
$p(f)$	内容块 $f \in F$ 的流行度
c_v	缓存节点 v 的最大容量 (内容块为单位)
r_v^f	内容块 $f \in F$ 在节点 $v \in V$ 处的请求速率
$b_{f,v}$	内容块 $f \in F$ 在节点 $v \in V$ 处缓存后的收益

考虑一个 NDN 区域由 $G(V, E)$ 描述，其中， $V = \{v_1, v_2, \dots, v_{|V|}\}$ 表示该区域内的路由器节点， E 定义了路由器节点之间的链路。路由器都带有缓存能力，且缓存容量为 c_i 。对于内容 $f_i \in F$ ，定义 $p(f_i)$ 为该内容的流行度，可知 $\sum_{f_i \in F} p(f_i) = 1$ 。这里，假设外部请求服从泊松到达。

根据上述描述将问题构造为

$$\text{Maximize: } \sum_{f \in F} \sum_v p(f) x_{f,v} b_{f,v}$$

Subject:

$$\begin{aligned} x_{f,v} &= \begin{cases} 0 \\ 1 \end{cases} \\ \sum_f x_{f,v} &\leq c_v \\ \sum_f p(f) &= 1 \end{aligned}$$

以上问题可描述为在给定网络拓扑及每个节点缓存容量及请求分布条件下，如何将内容合理放置到每个缓存节点，以获得最大化的收益。可以看出该问题属于多约束条件下的优化放置问题，为了解决该问题，需要网络拓扑、内容流行度、节点缓存信息等全局信息，然而这些在信息的采集、计算等方面都具有一定难度，会带来一定程度的开销。为了在较低开销情况下解决以上放置问题，本文提

出了一种轻量级的缓存协作机制——ECCS 策略，以下对 ECCS 策略做出详细介绍及性能评估。

4 ECCS 策略

4.1 ECCS 的基本思想

由于中间节点转发的内容请求是由其下游节点的请求汇聚而成，因此当某节点 R 的下游节点缓存了某块内容后，那么该节点 R 再次缓存该内容的价值就大大降低了。然而，以往的缓存策略往往是在数据返回路径上进行缓存决策，上游节点的缓存决策先于下游节点，因此，上游节点无法获知下游节点的缓存决定，从而会产生一些无效的缓存。本文的策略利用 NDN 拉模式(PULL)中内容请求路径与内容返回路径一致的特点，在数据请求过程中首先进行缓存决策，同时将下游节点的缓存决策及更新信息带给上游节点，帮助上游节点做出更加合理的缓存决策并且及时更新信息，充分发挥缓存的优势。

本策略具有“边缘优先”和“逐级反馈”这 2 个特点，以下对这 2 点进行详细阐述。

4.2 边缘优先

首先，由于边缘节点与请求者的距离更近，能够更好地感知用户请求的热度分布，而核心方向由于存在过滤效应等因素，对热度感知的速度和精度相比于边缘节点有较大差距。

此外，在大多数场合下，将热度高的内容放置于更加靠近请求者的节点以获得更高的收益。

考虑如图 1 所示的 2 种放置策略，黑色节点为内容源，定义某缓存节点缓存内容所获得的跳数收益为 h ，则在其下一层缓存内容获得的跳数收益为 $h+1$ ，定义 B_1 和 B_2 分别为如图 1 的 2 种内容放置方式的收益，以 $p(f)$ 表示内容块 f 的热度，容易求得

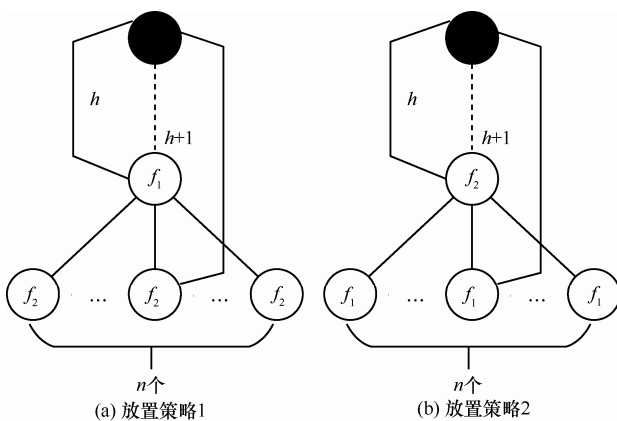


图 1 缓存放置策略

$$B_1 = p(f_1)h + \sum p(f_2)(h+1)$$

$$B_2 = p(f_2)h + \sum p(f_1)(h+1)$$

两式相减并整理得

$$B_2 - B_1 = [n + h(n-1)][p(f_1) - p(f_2)]$$

由于 $n \geq 1, h \geq 1 \Rightarrow n + h(n-1) > 0$

因此， $B_2 - B_1$ 的正负仅取决于 $p(f_1) - p(f_2)$ 。

由此可以推知流行度高的内容放置于下层节点以获得更高的收益。

基于以上 2 点考虑，本文将缓存决策的优先权交给边缘节点，即按照由边缘到核心的顺序依次进行缓存决策。详细策略描述将在 4.4 节叙述。

4.3 逐级反馈

根据边缘优先的特点可知，上游节点的缓存决策将在下游节点的缓存决策之后进行，一旦下游节点决定缓存某一内容块时，上游各个节点都将不再缓存该块。但是由于下游节点缓存某内容之后，上游节点请求分布等信息必然会产生一定的变化，因此，需要将请求信息的变化向上游转发，使上游节点的统计信息即时更新，辅助其完成之后的缓存决策。

为了确定向上游反馈信息的具体内容，进行以下分析。

与传统路由器不同，由于 NDN 路由器中存在网内缓存，请求将会经过缓存的过滤，如图 2 所示。因此，对于路由节点来说，内容请求的出速率将不再简单地等于各入速率之和，而是满足

$$outgoing = \sum_i incoming_i(1 - h)$$

其中， h 表示该路由节点的缓存命中率，细化至对象 k ，则节点请求输出速率 $outgoing_k$ 满足

$$outgoing_k = \sum_i incoming_{i,k}(1 - h_k)$$

其中， h_k 表示该节点对内容 k 的缓存命中率。

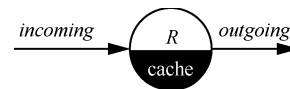


图 2 缓存过滤

以下分析都针对内容 k ，为了简化书写及方便阅读，以下各变量均省略了下标 k 。

以图 3 为例，节点 R 向上游节点转发对 k 请求的速率等于它的 3 个下游节点对 k 的请求速率之和乘以节点 R 的缓存未命中率。现假设节点 R 的下游节点 R_1 缓存了内容 k ，则在一定时间内节

点 R_1 及其下游各节点对 k 的请求将不会转发至 R ，因此，节点 R 收到的来自于 R_1 的对于 k 的请求速率将变为 0。同样，从 R 发出的对内容 k 的请求速率也随之变为了 $outgoing'$ ，可以求得该速率满足如下关系。

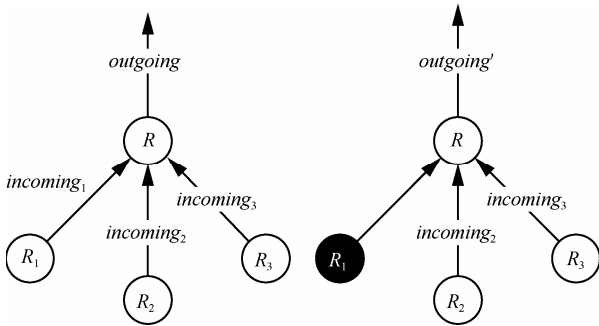


图 3 缓存过滤示例

$outgoing' = outgoing - \Delta outgoing$ ，其中

$$\Delta outgoing = incoming_1 h$$

又由于 $incoming_1 = outgoing_{R_1}$ ，以及

$$h = 1 - \frac{outgoing}{\sum_i incoming_i}$$

令 $incoming = \sum_i incoming_i$ ，可得

$$\Delta outgoing = outgoing_{R_1} \left(1 - \frac{outgoing}{incoming} \right)$$

其中， $\Delta outgoing$ 反映了下游节点缓存内容 k 给上游节点的请求速率带来的变化。通过以上计算可以将缓存 k 带来的请求速率的变化逐级地向上游节点反馈，实时更新上游节点的缓存决策信息，来帮助上游节点做出更加准确的缓存决策。

为了完成以上逐级反馈，需要统计节点的 $incoming$ 和 $outgoing$ ，同时需要在 Interest 分组中加入一个标记，将下游节点计算得出的 $\Delta outgoing$ 向上游转发，辅助上游节点完成后续的反馈。

通过综合考虑以上 2 个方面，本文设计了边缘优先逐级反馈的缓存协作策略。

为此在 NDN 分组中增加了 2 个字段，如图 4 所示。

Cache Node Identifier 用来标识该内容块将被缓存在哪个节点，该字段在 Interest 转发的过程中由决定缓存该内容块的节点改写。当其值被设置后，后续节点将不再进行缓存决策操作。在内容命中的节点将该字段转移至 Data 分组，并且在 Data 分组转发过程中通过匹配该字段与节点 ID 完成缓

存操作。

Cache Decision Information 在 Interest 转发过程中由决定缓存该内容块的节点及后续节点计算并改写，用来指导上游节点修正缓存统计信息。

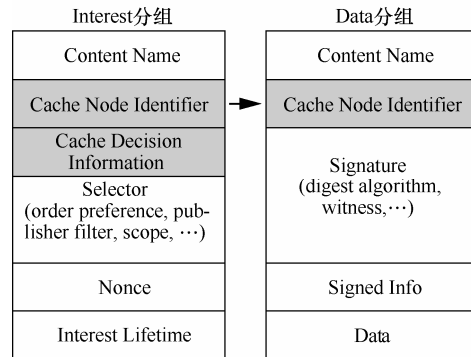


图 4 对 NDN 数据分组格式的修改

4.4 策略过程描述及示例

假设当前某块内容在请求者到内容源的路径上各节点均未缓存，此时请求者发出对该内容的请求。在本次请求过程中，会按照从请求者到内容源的顺序依次进行缓存决策（边缘优先），也就是说，由边缘节点首先进行缓存决策，假如该节点决定不缓存该内容，则再由该节点的下一跳节点进行决策，直到途中某个节点决定缓存该内容后，之后的节点不再进行缓存决策，而引发统计信息的更新过程。在决定缓存该内容的节点处，请求分组中的 Cache Node Identifier 字段被改写为该节点 ID，以便在内容返回时，通过匹配该字段保证内容在该节点缓存。同时修改 Cache Decision Information 字段，用来更新上游节点的统计信息。该节点后续各节点，都按照收到的请求分组中的 Cache Decision Information 字段，更新自己的热度统计信息，并且根据 4.3 节描述的计算方法求得对下一跳节点的热度信息的影响值，改写请求分组中的 Cache Decision Information，以完成逐级反馈。

如果在决策阶段没有节点决定缓存该内容，则返回的内容将只被转发到请求者，而不会被任何中间节点缓存。另外，如果请求的内容在中间节点命中时，会引发该内容的替换决策，这个问题将在 4.5 节进一步讨论。同样，对于决策依据以及决策中的特殊情况将在 4.5 节进行说明。

图 5 描述了一个请求路径上的缓存协作过程，节点 R_1 收到请求后，根据 Cache ID=NULL 可知其下游节点尚未决定缓存该内容，因此 R_1 启动缓存决

策判断是否要缓存内容。假设 R_1 决定不缓存该内容，则对缓存标记不做任何修改，继续转发请求。请求转发至 R_2 ，同样， R_2 需要进行缓存决策，假设 R_2 决定要缓存此内容，则修改 Cache ID 为本节点的 ID，同时计算并设置 Cache Info 的值，继续转发请求。此后的各节点收到请求后，通过检查 Cache ID 可以发现已有下游节点决定缓存此内容，则后续节点不再进行缓存决策，而根据 Cache Info 修改本地信息同时再次计算并修改 Cache Info 后向上游转发。当内容命中时，Cache ID 被转移到 Data 分组中。内容返回时，节点会将 Cache ID 与本身 ID 比

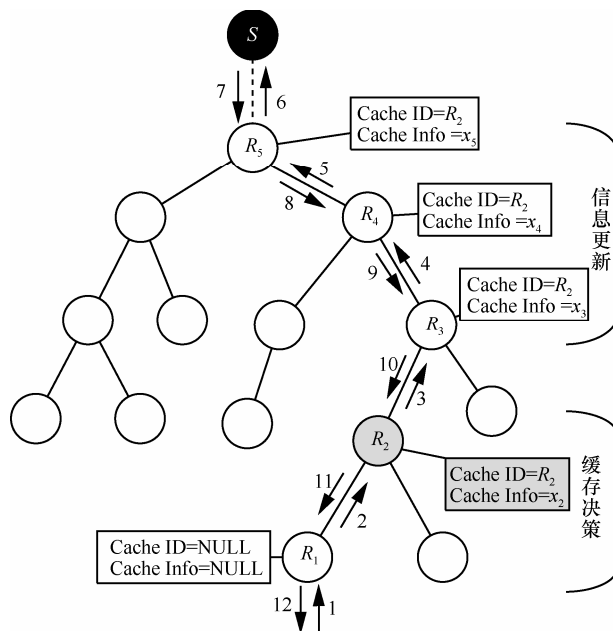


图 5 ECCS 策略示例

4.5 策略相关说明及算法描述

为了提高缓存空间利用率，当内容返回时，如果路径上某节点缓存尚有空闲，则直接缓存该内容，无需匹配 Cache ID。

进行内容插入操作时，如果缓存容量已满，则产生替换，可使用常用的替换算法，如 LRU。

当内容请求在中间节点缓存内命中时，如果存在某下游节点决定缓存该内容，则命中节点通过决策决定是否需要将该内容从缓存中删除。

本策略中需要做出缓存决策的情况如下。

1) 在 Interest 分组转发过程中，当本节点的所有下游节点都未决定要缓存该对象时，则本节点要进行缓存决策，确定本节点是否要缓存该内容块。决策的依据是该内容的统计热度值是否达到节点

要求的热度阈值 $threshold$ ，热度阈值则根据当前已缓存的内容热度值下限决定。

2) 当下游节点决定缓存某内容块，且在本节点缓存命中时，做出决策是否删除该内容块。删除决策的依据是该内容更新后的请求速率小于 $\alpha \cdot threshold$ ， $\alpha \in [0,1]$ 。表示当内容进入缓存之后，替换该内容的阈值将大于等于缓存进入的阈值，体现出缓存内容的粘滞性。当 α 取值为 0 时，表明当该节点将不再收到该内容的请求后才会将此内容删除。

3) 当下游节点决定请求该内容块，并且本节点缓存空间未满足时，决策是否缓存该内容块（目前实现中对于未满足的情况统一做缓存处理）。

结合以上说明给出本策略的算法 1、算法 2 和算法 3。

算法 1 Interest 分组转发过程中缓存决策

- 1) 节点收到 Interest 分组
- 2) 检查缓存信息
- 3) if (下游节点决定缓存)
- 4) 修改该内容的本地信息
- 5) 计算并更新缓存信息
- 6) if (该内容存在于缓存中)
- 7) 做出替换决策
- 8) 返回内容 return
- 9) else
- 10) 转发请求
- 11) else
- 12) 缓存决策
- 13) if (决定本地缓存)
- 14) 修改 Interest 中缓存信息
- 15) 转发 Interest

算法 2 Producer 处理过程

- 1) 读取 Interest 中的缓存节点标识信息
- 2) 在 Data 分组写入缓存节点标识信息

算法 3 Data 分组转发过程缓存决策

- 1) 节点收到 Data 分组
- 2) if (CS 未满足)
- 3) 缓存 Data
- 4) else
- 5) 读取 Data 分组中缓存节点标识信息
- 6) 将标识与自身标识匹配
- 7) if (匹配)
- 8) 缓存该内容块

9) 转发 Data 分组

5 仿真实验

为了评估 ECCS 的性能, 本文设计了仿真实验来验证该策略的具体效果。

5.1 实验设置

本文使用开源的仿真工具 NDNSim^[18]对以上策略进行了实现, 并且设计了仿真实验与目前主流的缓存策略进行对比。NDNSim 是基于 NS-3^[19]的仿真工具, 加入了 NDN 的协议栈, 可以方便地进行 NDN 相关的仿真实验。本文修改了 NDNSim 中的分组结构, 增加了完成策略所需的字段, 并且在转发逻辑中增加了部分操作, 用以实现 ECCS 策略。

实验分别采用 BA (Barabási-Albert)模型^[20]和 WS (Watts-Strogatz)模型^[21]构建符合域内和域间特性的网络拓扑, 节点数为 100, 其中包含 66 个边缘路由器, 即产生外生请求的节点、33 个核心路由器和 1 个内容源节点。每个路由器上包含一定数量的缓存并且运行相应的缓存策略。用户的请求分布服从 Zipf 分布^[22], 请求到达符合泊松过程。实验仿真进行 180 s, 前 60 s 为预热时间, 取后 120 s 的数据作为结果分析数据。其他参数参如表 2 所示。

表 2 实验参数

参数	默认值	变化范围
Zip 分布参数	1.0	0.7~1.3
内容总数	50 000	20 000~80 000
缓存容量/slot	40	10~100
请求速率/(request·s ⁻¹)	200	50~350

5.2 对比方案

为了验证 ECCS 策略的性能, 本文选择了以下几种主流的缓存策略与其对比, 替换算法采用 LRU 算法。

CEE (cache everything everywhere)^[2]: 传统的内容随处缓存方案。

LCD (leave copy down)^[6]: 内容只在命中节点下一跳节点存储。

MCD (move copy down)^[6]: 内容只在命中节点下一跳节点存储, 并同时删除原内容。

Probcache^[8]: 内容放置的概率与缓存收益以及内容存储能力成正比。

MBP^[15]: 基于内容热度和位置信息确定缓存概率。

5.3 性能评价指标

为了客观地反映不同缓存策略的实际效果及不同参数对缓存性能的影响, 本文定义了以下性能评价指标。

服务负载减少率(SLRR, server load reduction ratio)为

$$SLRR = \frac{S_counts}{R_counts}$$

其中, S_counts 表示服务器响应请求的次数, R_counts 表示用户请求总次数。

服务负载减少率反映由于缓存响应用户请求带来的服务器端的负载压力的减少比率。该指标越高, 表示服务器端负载压力越小, 同时说明缓存起到的效果越明显。

平均跳数减少率(AHRR, average hop reduction ratio)为

$$AHRR = 1 - \frac{\sum_{i=1}^{R_counts} hops_c}{\sum_{i=1}^{R_counts} hops_nc}$$

其中, $hops_c$ 表示使用某种缓存策略后用户请求经过的跳数, $hops_nc$ 表示当不使用缓存时用户请求经过的跳数, R_counts 表示用户请求总次数。

平均跳数减少率描述了加入缓存系统后用户响应速度的改善程度, 是反映用户体验水平的重要指标。

平均缓存命中率(ACHR, average cache hit ratio)为

$$ACHR = \frac{\sum_r Hit_counts_r}{\sum_r R_counts_r}$$

其中, R_counts_r 表示路由器 r 收到的请求数, Hit_counts_r 表示在路由器 r 命中的请求数量。

该值反映了平均的缓存节点请求命中率, 该值越大表示缓存效果越佳。

缓存冗余度 (CR, cache redundancy) 为

$$CR = \frac{Cached_counts}{Type_counts}$$

其中, $Cached_counts$ 表示缓存的内容数量, $Type_counts$ 表示缓存的内容种类。

该指标反映了相同内容在缓存中存储的平均次数，也就是内容在缓存中的冗余度，该值越小表明冗余度越低，当取最小值 1 时，表明当前缓存的内容不重复。

缓存内容差异率(CDR, content diversity ratio)为

$$CDR = \frac{Type_counts}{Cache_size}$$

其中, *Type_counts* 表示缓存的内容种类, *Cache_size* 表示缓存容量 (以 slot 为单位)。

该指标反映了一条缓存路径上缓存内容的种类差异, 该值越大表示缓存中的内容种类越多, 内容差异性越明显, 当该指标取最大值为 1 时表明当前缓存的内容都是不重复的。该指标与 *CR* 的关系是: 当缓存空间全部被利用时, $CDR = \frac{1}{CR}$ 。

链路负载率(LTR, link traffic rate)记录了在仿真过程中, 排名链路负载前 10 位的即时链路负载值, 该值反映出不同缓存策略对链路负载压力缓解的效果。

5.4 实验结果

为了观察网络性能受某个参数的影响, 一次只有一个参数变化, 而其他参数保持不变, 其取值如表 2 所示。

5.4.1 Zipf 分布参数的影响

目前, 研究者发现用户的内容请求行为服从 Zipf 分布^[22], 在不同的场景下 Zipf 分布的参数具有一定差异性, 本文通过改变 Zipf 分布的参数 (0.7~1.3) 来研究其对缓存性能带来的影响, 同时, 比较不同缓存策略的实际表现。

图 6 给出实验结果, 可以看出随着 Zipf 参数的增大, 各种缓存策略在服务负载减少率、平均跳数减少率和平均缓存命中率都有不同程度改善, 这主要归因于 Zipf 参数的增大使内容请求的局部性特性更加明显, 更有利于缓存系统发挥作用。同时, 可以看出 ECCS 策略相比于其他策略具有明显优势, 说明通过 ECCS 使缓存的性能得到进一步提升。

5.4.2 缓存容量的影响

通过对节点缓存容量的改变 (10~100 slot), 考察缓存性能的改变。观察图 7 可以发现, 缓存容量从 10 增大到 100 的过程中, 大部分策略的缓存效果逐渐改善, 说明增大缓存容量能够对提高缓存性能起到一定作用。但同时可以发现缓存性能曲线

上凸, 表明缓存性能改善的速率呈下降趋势。ECCS 在各策略中仍具有最好表现。另外, 可以发现, MCD 策略在缓存容量增大过程中并没有表现出明显的性能改善, 这主要是由于 MCD 策略本身对于缓存空间的利用率较低, 本文还将讨论 MCD 缓存策略的低效问题。

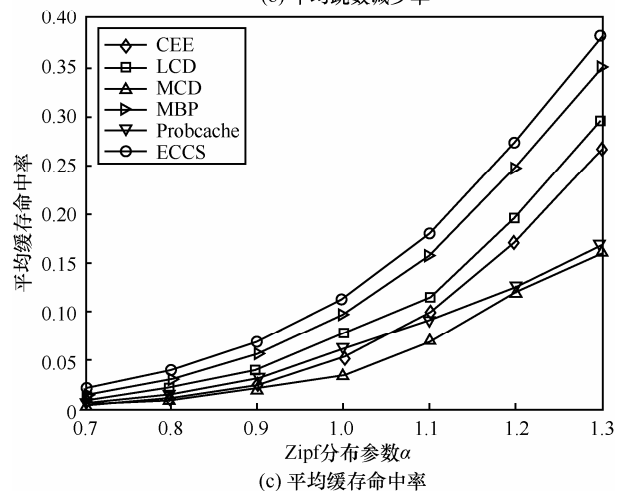
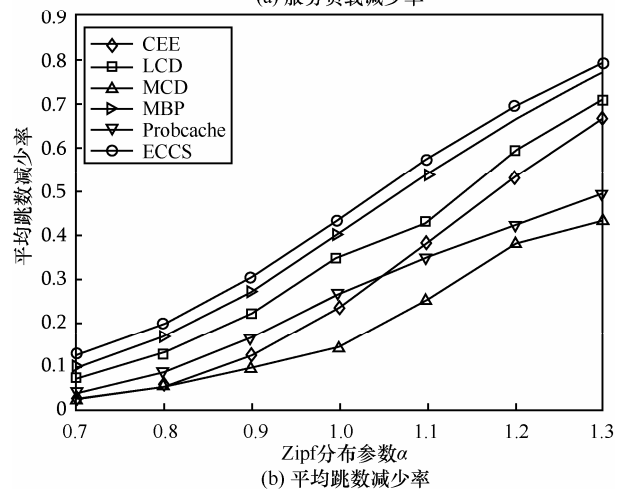
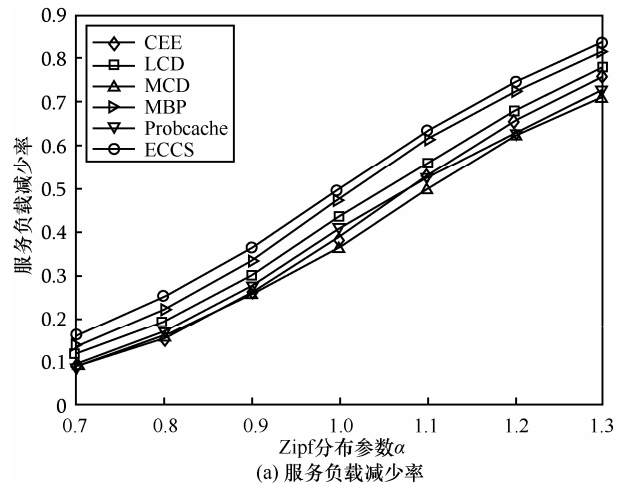


图 6 Zipf 分布参数的影响

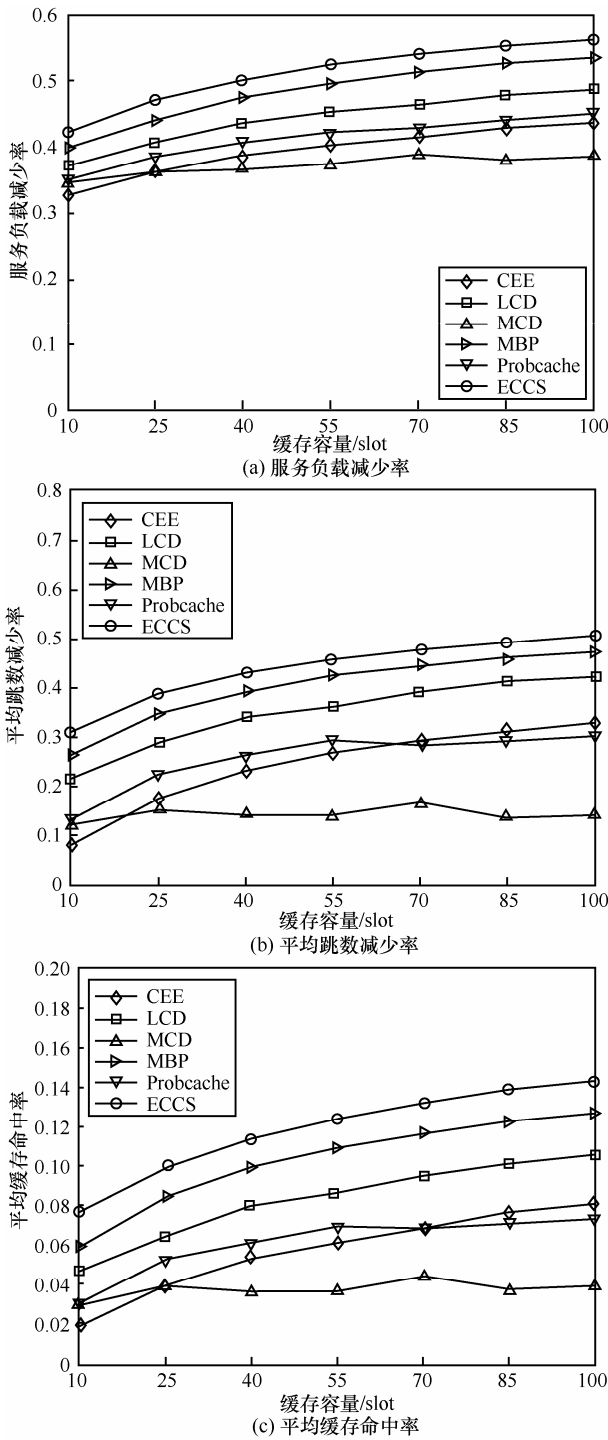


图 7 缓存容量的影响

5.4.3 内容总量的影响

本文通过改变内容的总量 (20 000~80 000 slot) 来观察内容总量对缓存性能的影响。由图 8 看出, 内容总量由 20 000 增加至 80 000 的过程中, 图 8 中 3 项指标都呈下降趋势。这是由于在其他参数不变的情况下, 内容总数的增加使内容的局部性特性相对减弱, 用户的请求更为分散, 因此缓存的效果也

逐渐降低。同时 ECCS 仍然在各种策略中具有最优表现。

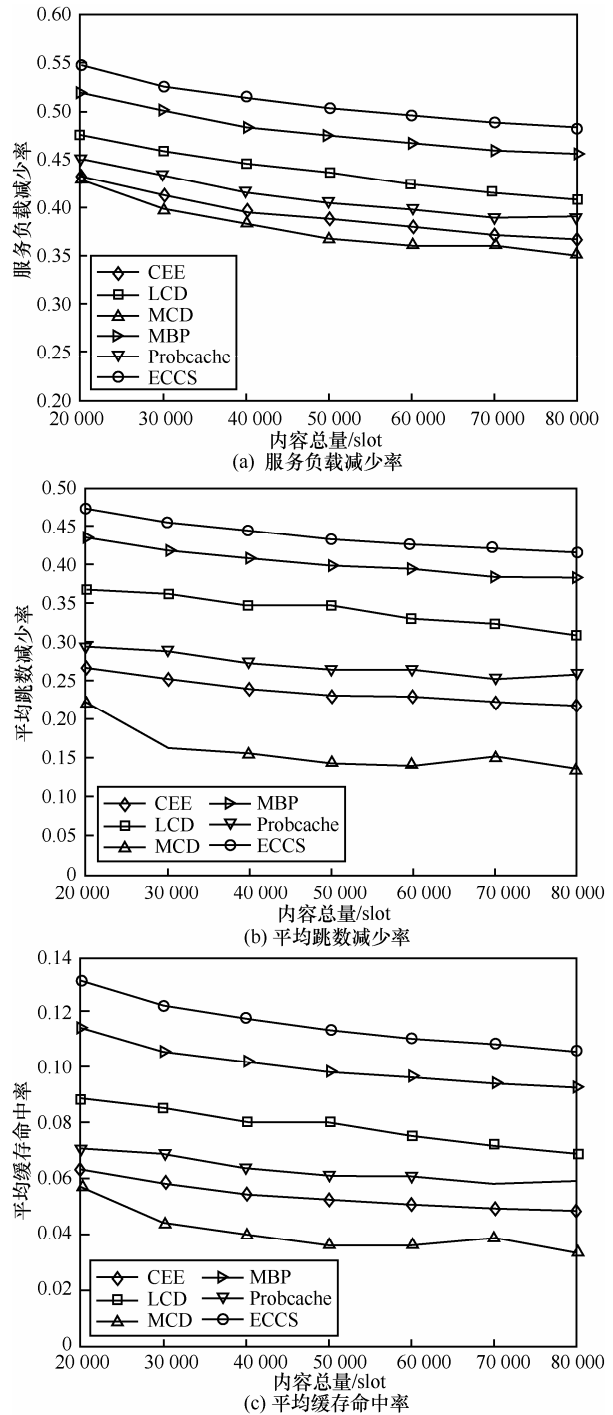


图 8 内容总量的影响

5.4.4 请求速率的影响

本文通过改变用户请求速率 (50~350 request/s) 来观察不同速率下缓存性能的变化。通过图 9 可以看出, 在不同的请求速率下, 缓存的评价指标没有明显的变化趋势, 因此, 尚不能说请求速率的改变

对缓存效果有明显的影 响，但 ECCS 策略相比于其 他策略仍具有明显优势。

的结果都是在单条传输路径上统计得出的。通过实 验发现以下情况。

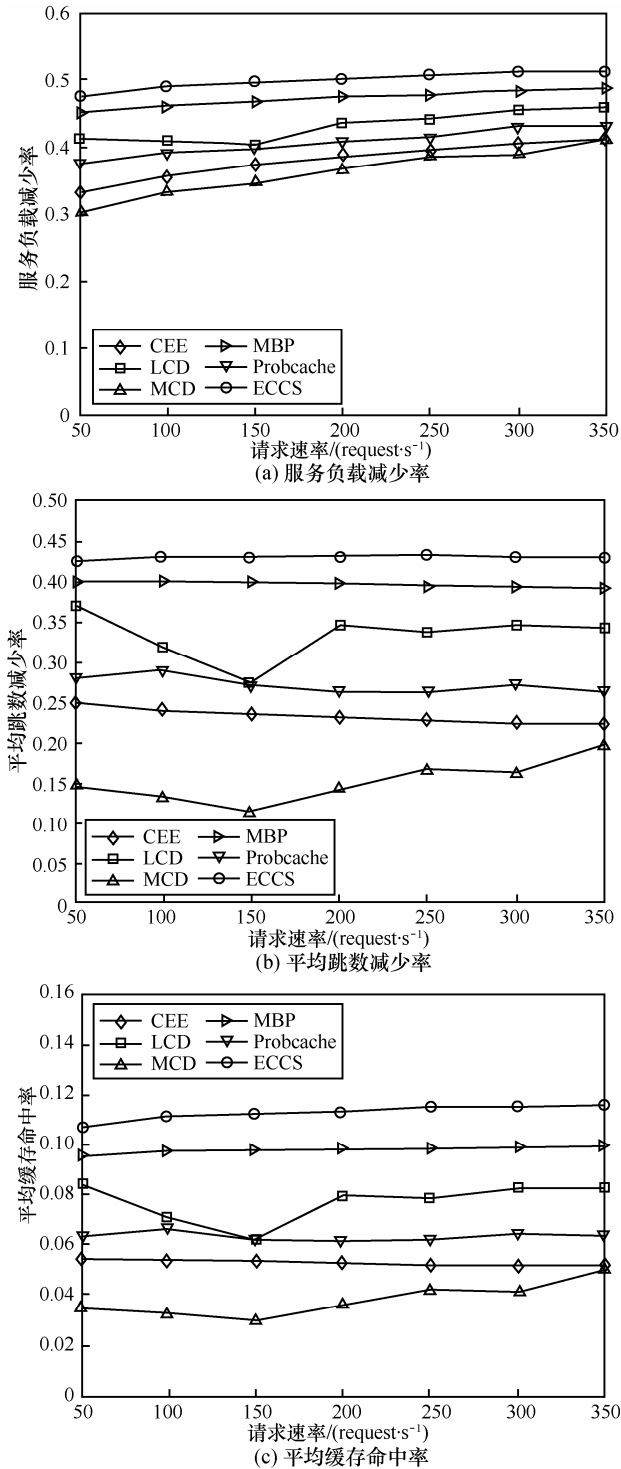


图 9 请求速率的影响

5.4.5 冗余度、多样性与缓存性能的讨论

图 10 和图 11 讨论了缓存冗余度和缓存内容差异性在不同指标变化下的具体表现。以上 2 个指标

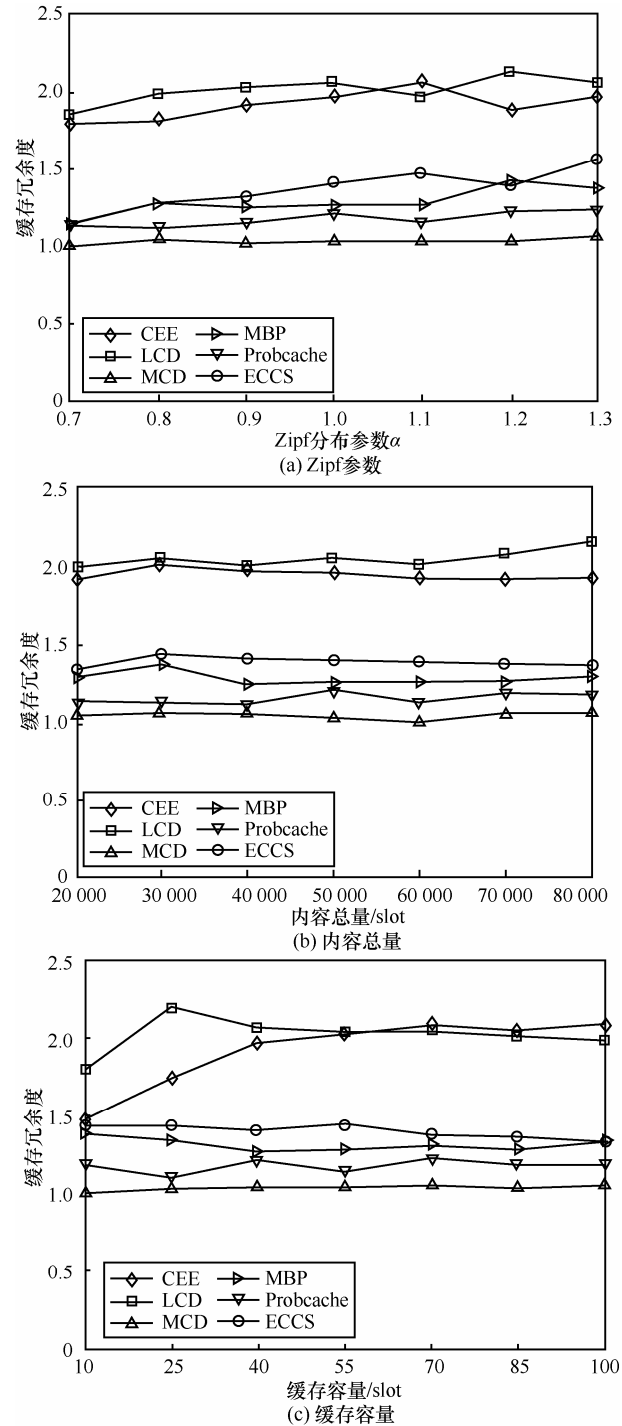


图 10 缓存冗余度

LCD 策略在缓存冗余度和内容多样性方面并没有表现出预期的效果，而 CEE 也并没有之前人们认为的很差的表现。这两者在以上 2 个指标上表现相近。这主要是由于在复杂网络拓扑中，一条传输

路径上的节点同时也存在于其他传输路径中。在大量用户同时请求过程中，由于多条传输路径的叠加使 CEE 产生的缓存冗余大大降低。如实验数据是在一条由 12 个节点组成的传输路径中产生，而 CEE 的缓存冗余度只有 2.0 左右，即平均来说，同样的内容在该路径上只存在 2 个副本，这与之前认为的 CEE 带来的高冗余的缺陷有所不同。

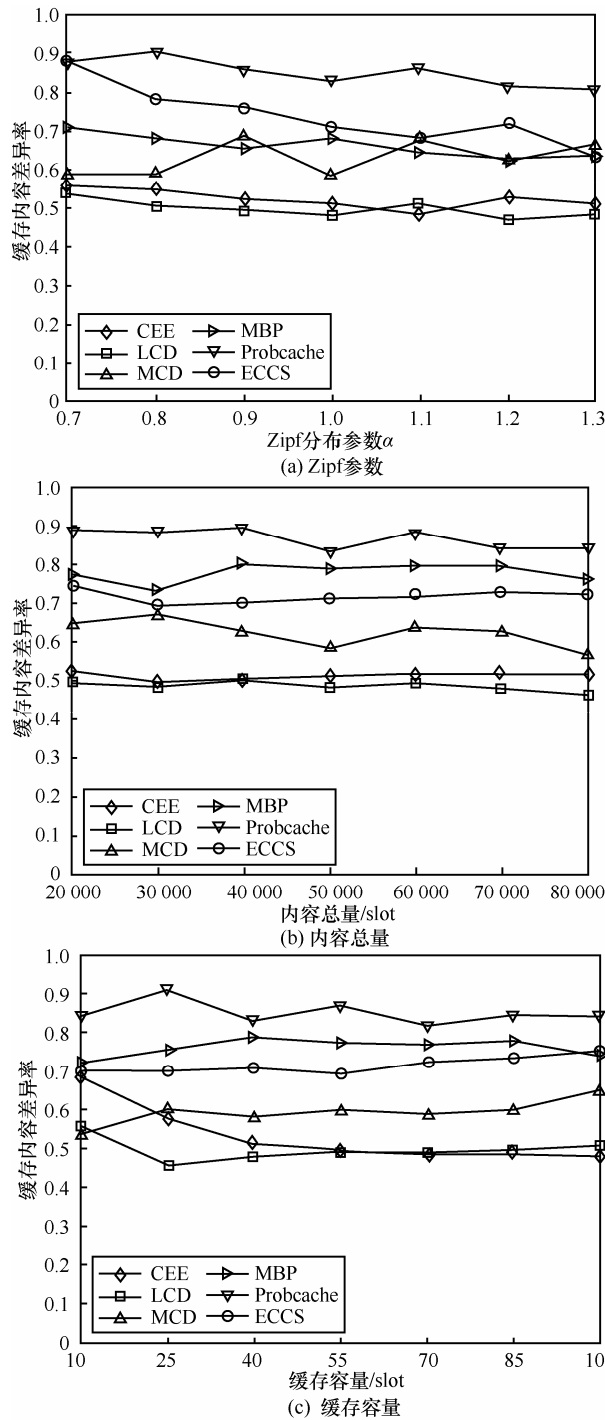


图 11 缓存内容差异率

MCD 在实验结果中冗余度接近于 1，然而从之前的部分可以看出 MCD 的缓存效果并不好。本文通过分析数据发现，MCD 由于在每次缓存命中之后必然将内容向下缓存一跳，同时删除本地副本，这样带来的结果是大量缓存空间闲置，无法充分利用所有的缓存容量。数据表明 MCD 策略只利用了缓存容量的 85%左右，而剩余的缓存空间由于替换操作而长期空闲，无法发挥应有的作用。如果把空闲的缓存条目也当作内容来统计，则 MCD 的缓存冗余度高达 12 左右，这也解释了为什么 MCD 在其他缓存性能指标上表现不佳。

Probcache 在这 2 项指标上都有较好表现，这主要由于 Probcache 是一种概率缓存机制，同时没有对内容本身进行区分，使不同热度的内容块在其他条件相同的情况下有同等的缓存概率，使缓存中内容的分布更加平均。但这样也会使一些冷门的内容缓存下来，因此其他缓存性能指标并不突出。

通过以上结果发现，复杂网络条件下的内容冗余与线性拓扑具有很大区别，冗余度与缓存性能之间没有绝对的对立关系，适当的内容冗余有可能会带来整体缓存性能的提高。

5.4.6 链路负载缓解

图 12 显示了各种策略在链路负载方面的表现。图 12 显示了 10 条最高负载的链路负载数据，可以看出相比于没有缓存的情况，使用缓存策略在链路负载率上都有不同程度降低，而其中 ECCS 的负载降低最为明显，表明了使用 ECCS 策略对于改善链路负载率同样具有良好的效果。

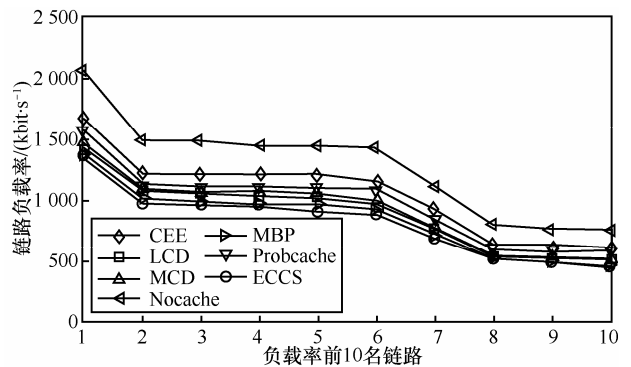


图 12 链路负载率

通过以上各组实验发现，相比于其他策略，ECCS 的性能在以上评价指标中具有明显优势。对比于 CEE，ECCS 在服务负载减少率、平均跳数减少率、平均缓存命中率方面，分别提高了 32.31%、

104.56%、132.99%。

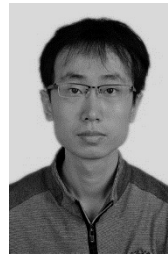
6 结束语

为了进一步发挥 ICN 网内缓存的潜在性能, 本文提出一种边缘优先逐级反馈的缓存协作策略 (ECCS)。通过在请求转发过程中提前进行缓存决策, 将下游节点的缓存决策信息及内容统计信息告知上游节点, 辅助上游节点完成协作缓存。实验结果表明, ECCS 在服务负载减少率、平均跳数减少率、平均缓存命中率等方面与现有方法相比有显著优势。

参考文献:

- [1] ZHANG G, LI Y, LIN T. Caching in information centric networking: a survey[J]. Computer Networks the International Journal of Computer & Telecommunications Networking, 2013, 57(16): 3128-3141.
- [2] JACOBSON V, SMETTERS D K, THORNTON J D, et al. Networking named content[C]//The ACM CoNEXT. 2009: 1-12.
- [3] TEEMU K, MOHIT C, CHUN B G, et al. A data-oriented (and beyond) network architecture[J]. ACM SIGCOMM Computer Communication Review, 2007, 37(4):181-192.
- [4] DANNEWITZ C, GOLIC J, OHLMAN B, et al. Secure naming for a network of information[C]//The IEEE INFOCOM Workshops. 2010: 1-6.
- [5] NIKOS F, PEKKA N, DIRK T, et al. Developing information networking further: from PSIRP to PURSUIT[J]. Broadband Communications, Networks and Systems, 2012, 66:1-13.
- [6] LAOUTARIS N, CHE H, STAVRAKAKIS I. The LCD interconnection of LRU caches and its analysis[J]. Performance Evaluation, 2006, 63(7):609-634.
- [7] CHAI W K, HE D, PSARAS I, et al. Cache less for more in information-centric networks[C]//The Networking 2012. 2012: 27-40.
- [8] PSARAS I, CHAI W K, PAVLOU G. Probabilistic in-network caching for information-centric networks[C]//The Second Edition of the ICN Workshop on Information-Centric Networking. 2012: 55-60.
- [9] 葛国栋, 郭云飞, 兰巨龙, 等. CCN 中基于替换率的缓存空间动态借调机制[J]. 通信学报, 2015, 36(5): 124-133.
GE G D, GUO Y F, LAN J L, et al. Dynamic cache size transfer scheme based on replacement rate in content centric networking[J]. Journal on Communications, 2015, 36(5): 124-133.
- [10] WANG S, BI J, WU J, et al. Could in-network caching benefit information-centric networking[C]//AINTEC. 2011: 112-115.
- [11] 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.
LIU W X, YU S Z, HU X, et al. Selective caching in content-centric networking[J]. Chinese Journal of Computers, 2014, 37(2): 275-288.
- [12] MING Z, XU M, WANG D. Age-based cooperative caching in information-centric networks[C]//IEEE INFOCOM 2012 on Computer Communications Workshops. IEEE, 2012: 268-273.
- [13] CHO K, LEE M, PARK K, et al. WAVE: popularity-based and collaborative in-network caching for content-oriented networks[C]//The IEEE INFOCOM Workshop on NOMEN. 2012: 316-321.
- [14] REN J, QI W, et al. MAGIC: a distributed max-gain in-network caching strategy in information-centric networks[C]//Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference. 2014: 470-475.
- [15] WU H, LI J, ZHI J. MBP: a max-benefit probability-based caching strategy in information-centric networking[C]// IEEE International Conference on Communications. 2015: 5646-5651.
- [16] WANG L, HOQUE A, YI C A, et al. OSPFN: an OSPF based routing protocol for named data networking[R]. University of Memphis and University of Arizona, Tech Rep, 2012.
- [17] AKM M H, SYED O A, ADAM A, et al. NLSR: named-data link state routing protocol[C]//The ACM SIGCOMM Workshop. 2013: 15-20.
- [18] AFANASYEV A, MOISEENKO I, ZHANG L. ndnSIM: NDN simulator for NS-3, NDN[R]. Technical Report NDN-0005, 2012.
- [19] HENDERSON T R, et al. Ns-3 project goals[C]//The 2006 Workshop on NS-2: the IP Network Simulator. 2006.
- [20] BARAB A, ALBERT A R. Emergence of scaling in random networks[J]. Science, 1999, 286(5439): 509-512.
- [21] WATTS D J, STROGATZ S H. Collective dynamics of small world networks[J]. Letters to Nature, 1998, 393: 440-442.
- [22] LEE B, PEI C, LI F, et al. Web caching and Zipf-like distributions: evidence and implications[C]//The IEEE INFOCOM. 1999:126-134.

作者简介:



智江 (1983-), 男, 山西太原人, 中国科学院大学博士生, 主要研究方向为下一代互联网关键技术、缓存技术。

李俊 (1968-), 男, 安徽桐城人, 中国科学院计算机网络信息中心副总工、博士生导师, 主要研究方向为未来互联网、网络安全等。

吴海博 (1981-), 男, 山东新泰人, 博士, 中国科学院计算机网络信息中心助理研究员, 主要研究方向为未来互联网、缓存技术、对等网络。

任勇毛 (1981-), 男, 湖南邵阳人, 博士, 中国科学院计算机网络信息中心副研究员, 主要研究方向为未来互联网、传输协议、拥塞控制等。