基于节点介数和替换率的内容中心网络网内缓存策略

崔现东*[①] 刘 江[①] 黄 韬[①] 陈建亚[②] 刘韵洁[①③]
[①](北京邮电大学泛网无线通信教育部重点实验室 北京 100876)
[②](北京邮电大学北京市网络体系构建与融合重点实验室 北京 100876)
[③](南京(中国)未来网络产业创新中心 南京 211100)

摘 要：网内缓存技术是内容中心网络(CCN)的关键技术之一，CCN 采用传统的 ALWAYS 缓存策略，会造成较大冗余。改进的 Betw 方案仅考虑了节点介数，容易造成高介数节点缓存更替频繁，内容可用性下降。为了解决这个问题，该文提出一种综合使用网络节点介数和节点缓存内容更替速率作为缓存决策度量的新型网内缓存策略 BetwRep，通过权衡节点位置重要性和缓存内容时效性实现回传内容的最佳放置。最后，基于 ndnSIM 平台进行的网络仿真表明，该文提出的 BetwRep 缓存策略取得了比 Betw 方案和 ALWAYS 方案更低的源端请求负载和更少的平均跳数。

关键词：内容中心网络；网内缓存技术；BetwRep 缓存策略；节点介数；缓存替换率
中图分类号：TP393 **文献标识码**： A **文章编号**：1009-5896(2014)01-0001-07
DOI: 10.3724/SP.J.1146.2013.00503

A Novel In-network Caching Scheme Based on Betweenness and Replacement Rate in Content Centric Networking

Cui Xian-dong[①] Liu Jiang[①] Huang Tao[①] Chen Jian-ya[②] Liu Yun-jie[①③]
[①](*Key Laboratory of Universal Wireless Communications*, *Ministry of Education*, *Beijing University of Posts and Telecommunications*, *Beijing* 100876, *China*)
[②](*Beijing Key Laboratory of Network System Architecture and Convergence*, *Beijing University of Posts and Telecommunications*, *Beijing* 100876, *China*)
[③](*Future Network Industry Innovation Center of China*, *Nanjing* 211100, *China*)

Abstract: In-network caching is one of the key aspects of Content Centric Networking (CCN), which is widely concerned recently. However, the ALWAYS caching scheme (caching everywhere on the delivery path) in CCN produces a great of redundancy, while the Betw scheme leads to that the node has the more frequent replacement with the larger betweenness centrality, which will decrease the availability of the content. In this paper, a novel in-network caching scheme named BetwRep is proposed based on a metric including the betweenness centrality and the replacement rate of one node to address the problem-where to cache along the delivery path. Simulation experiment based on ndnSIM demonstrates that the BetwRep caching scheme achieves the lower loading in the source server and less average hops than that of Betw scheme and ALWAYS scheme.

Key words: Content Centric Networking (CCN); In-network caching; BetwRep caching scheme; Betweenness centrality; Replacement rate

1 引言

内容中心网络(CCN)是一个基于内容的通信架构，每个节点都有一定的存储功能，用户会对所请求的内容发送兴趣包，当兴趣包在网络中某一个节点的缓存里找到请求内容后，内容将作为响应沿着兴趣包传播的路径原路返回，此时称为请求在节点缓存命中[1,2]。网内缓存技术在 CCN 中扮演着重要的角色，缓存策略的好坏对于 CCN 的性能具有较大的影响。缓存的研究主要有两个方面：一是单个节点缓存内容的替换策略，另一个是判定内容是否被节点缓存的判决策略[3]。单节点缓存策略已经得到了深入的研究，例如 LRU 和 LFU[4−7]等，不过单节点缓存替换策略只考虑一个节点的输入输出，忽略了网络拓扑对缓存性能的影响。网内缓存决策技术源于 web 缓存、CDN 和 P2P 系统，主要包括

2013-04-16 收到，2013-07-29 改回
国家973计划项目(2012CB315801, 2011CB302901)和中央高校基本科研业务费专项资金(2013RC0113)资助课题
*通信作者：崔现东 cuixdbupt@gmail.com

LCE(Leave Copy Everywhere), LCD (Leave Copy Down), MCD(Move Copy Down)[8]等。LCE 方法要求返回内容在分发路径上所有节点都要缓存，这样会导致网络中出现大量内容冗余备份，降低内容的多样性，本文将用到 CCN 中的传统 LCE 策略称之为 ALWAYS 策略。而 LCD 请求内容只在其命中点下一跳节点缓存，这使得流行度高的内容需要被访问多次才能缓存到边缘节点，同样会产生大量的内容冗余备份。MCD 类似于 LCD，不同之处是需要删除内容命中点的缓存(命中节点是内容源端除外)，以减少用户到内容源端路径上的重复备份，但内容缓存点的动态性会产生更多的网络开销。文献[9,10]基于信息中心网络(ICN)架构提出了 Betw(又或 Betw+LRU)策略，内容在返回时选择兴趣包请求路径上最重要的节点缓存，其它节点不再缓存。对于不同网络拓扑，该策略都取得了较高的网内节点缓存命中率，并减少了内容传输的平均跳数。然而，在实际网络中，节点缓存量远小于内容总量，Betw 策略会导致节点越重要，到达的请求越多，需要缓存的内容也越多，同时节点负载也会越大，从而导致缓存中的内容更替频繁，新缓存的内容，即使具有很高的流行度，也具有较大可能性被快速替换掉，致使后续请求无法充分利用前期缓存。

基于以上方案存在的问题，本文提出了一种综合使用网络节点介数和节点缓存内容更替率作为缓存决策度量的新型网内缓存策略 BetwRep。在返回内容判决路径上哪些节点需要缓存该内容时，既考虑节点的重要性，又要考虑节点缓存内容更替状况。该策略既有效保证了内容尽量缓存在相对重要的节点上，又能通过节点的内容替换率来调控内容的缓存，使重要节点避免处于高频率的内容替换状态而导致系统性能下降。

文章组织结构如下。第 2 节介绍 CCN 路径缓存模型以及相关技术；第 3 节重点阐述了本文提出的 BetwRep 缓存判决策略；第 4 节详细描述了仿真环境、方案和参数的配置，并对仿真结果进行分析；第 5 节是结束语。

2 CCN 路径缓存模型和相关技术

2.1 系统模型

信息中心网络研究文献中[11]，均假设 ICN 架构已经具有了一个发布/订阅机制功能模块，因此在建立系统模型时，不用再去考虑内容请求和解析机制。本文的系统模型假设 $G=(V,E)$ 是一个无向图，N 个节点表示为 $V=\{v_1,\cdots,v_N\}$，M 条边表示为 $E=\{e_1,\cdots,e_M\}$，$F=\{f_1,\cdots,f_K\}$ 是系统中的内容，$S=\{s_1,\cdots,s_P\}$ 是内容源端(内容服务器)的集合，内容初始随机分布在源端，请求来自于系统外部，到达服从泊松分布，到达率是 $\lambda=\sum_{r=1}^{K}\lambda_r$，$\lambda_r$ 是访问 f_r 的内容请求率。缓存命中是指一个请求在内容分发路径上某个节点找到请求内容，反之称为缓存未命中。如果缓存未命中，内容请求将遍历内容分发路径直到内容源端。按照已有文献的通用做法，假设内容单元大小相同，当节点的缓存已满，将采用 LRU 替换策略，新到达的内容替换掉缓存中最近最少使用的内容。

2.2 ALWAYS 与 Betw 路径缓存策略分析

有关兴趣包在节点(包括内容源端)缓存命中后，沿着内容返回的路径 L 上选择节点缓存内容的问题，现有CCN文献中大都采用传统的ALWAYS缓存策略，要求返回内容在分发路径上所有节点都要缓存，这样会导致网络中出现大量内容冗余备份，降低内容的多样性[10]。

文献[10]提出的Betw缓存策略，请求命中后将返回内容只缓存在分发路径最重要的节点上。由于在重要节点上，同一个内容被请求的次数会更多，使得内容在节点被缓存时间变长，这在某种程度上减缓了节点的缓存更替压力。而节点重要程度以介数作为度量，介数是节点介数中心性的简称[12,13]，用来描述节点在网络中重要性的一个度量，源于复杂网络领域，其数学定义如下：

假设 $G=(V,E)$ 是一个无向图，具有 n 个顶点，那么节点 v 的介数 $C_{B-SP}(v)$ 由式(1)给出，

$$C_{B-SP}(v)=\sum_{s\neq v\neq t\in V}\frac{\sigma_{st}(v)}{\sigma_{st}} \tag{1}$$

其中 σ_{st} 是顶点 s 到顶点 t 的所有最短路径数，$\sigma_{st}(v)$ 是其中经过顶点 v 的最短路径数。介数的计算有两种方法，分别是基于最短路径的和基于流的方法，考虑到网络中路由选路采用最短路径，选择前者。介数越大，节点在网络中越重要，意味着将控制更多的流(包括信息、资源等)。

然而，网络中节点缓存大小与系统内容总量相比非常小[14]，因此，重要节点到达的请求和经过的内容量都极为庞大，最流行的内容也会出现很快就被替换的情况，此时大量的兴趣包到达重要节点后，无法实现缓存命中，兴趣包也只能被转发到别的节点，直至内容源端。

下面通过一个实例来分析 ALWAYS 与 Betw 两种缓存策略存在的问题。考虑图 1(a)中的拓扑，假设 t=0 时刻，所有节点的缓存存储(CS)为空，用户 c_1 请求源 S 上的内容 f，当请求到达 S 后，内容将会

沿着请求的路径原路返回到 c_1，即经过 $v_1 \rightarrow v_2 \rightarrow v_4 \rightarrow v_6$。对于 ALWAYS 缓存策略，路径上 4 个节点都将会保留一份内容副本。假设现在又有用户 c_2 请求内容 f，如果采用 ALWAYS 缓存策略，请求在 v_4 获得内容，实现缓存命中，但是缓存在节点 v_1,v_2,v_6 的 f 成了冗余内容。根据 Betw 策略，如图 1(b)所示，在由 c_1 发出的兴趣包请求内容的路径上，节点 v_2 的介数最大，内容会被缓存在该节点，由此来自用户 c_2 的请求，经过较少的跳数获取内容，实现本地节点缓存命中，同时还不会给系统带来冗余内容。但当节点缓存远小于内容总量时，作为最重要的节点，大量兴趣包经过 v_2，缓存中的内容更替也会更频繁，以至于流行度高的内容也会被新内容快速替换掉。如图 1(c)所示，被用户 c_1 请求的内容 f 缓存在 v_2，随后 c_2 发出请求 f 的兴趣包，但 v_2 在短时间内已经将 f 更替掉，最后请求兴趣包只能从内容源端 S 处获取内容。这使得用户获取内容将经过更多的跳数，导致节点内容缓存更替频繁，形成恶性循环，使得 CCN 网络网内缓存性能下降。

3　基于节点介数和缓存更替率的 BetwRep 缓存策略

3.1 BetwRep 缓存策略核心思想

CCN 网络中如果一个节点位于多个内容分发路径上，便会有更多的兴趣包和内容包经过，则在该节点上更容易发生内容请求缓存命中。文献[10]提出的内容缓存算法选择介数作为度量，将返回内容只缓存在路径最重要的节点上。由于在重要节点上，同一个内容被请求的次数会更多，使得内容在节点被缓存时间变长，这在某种程度上减缓了节点的缓存更替压力。

本文目标是设计能取得高收益的网内缓存策略，并验证系统在不同缓存策略下的网络缓存性能。Betw 缓存策略用来判决请求内容在返回路径上如何缓存，但由于 Betw 策略选取缓存节点只考虑节点的重要性，而网络中节点缓存大小远小于系统内容总量[14]，大量的请求和内容到达重要节点，导致重要节点内容替换频繁，使得流行内容在节点中的缓存时间变短，最后致使网内缓存命中率下降。另一个方面，作为网络层的技术，CCN 的节点内容缓存和数据转发要求达到线性处理速度，如果出现上述情况，节点处理包的压力将会陡增，给网络带来拥塞。

通过以上分析可知，系统中采用 Betw 缓存策略，改变了网络原有的缓存特性，导致网络性能参数发生改变，比如链路流量分布、节点负载分布等等。把节点介数作为衡量节点重要性的唯一标准，将返回内容只缓存在以此选出的重要节点上，无法避免网络参数改变给网络性能带来的负面影响。针对上述问题，本文将综合使用网络节点介数和描述节点缓存状态的参量作为缓存决策度量，设计新型网内缓存策略，通过权衡节点位置重要性和节点状态实现回传内容的最佳放置。由于网络达到稳定状态后，大多数节点缓存空间是满的，因此无法用节点缓存资源剩余量来衡量节点的负载和缓存状态，为此我们引入一个重要的判决参数 —— 节点的内容缓存替换率。作为描述缓存内容替换频率的参量，内容替换率可以准确地表征出节点的负载和缓存状态，并能有效地体现出节点内容的时效性。

本文利用节点的介数和内容缓存替换率这两个参数，设计一个新的节点度量 $M(v)$，以此值的大小

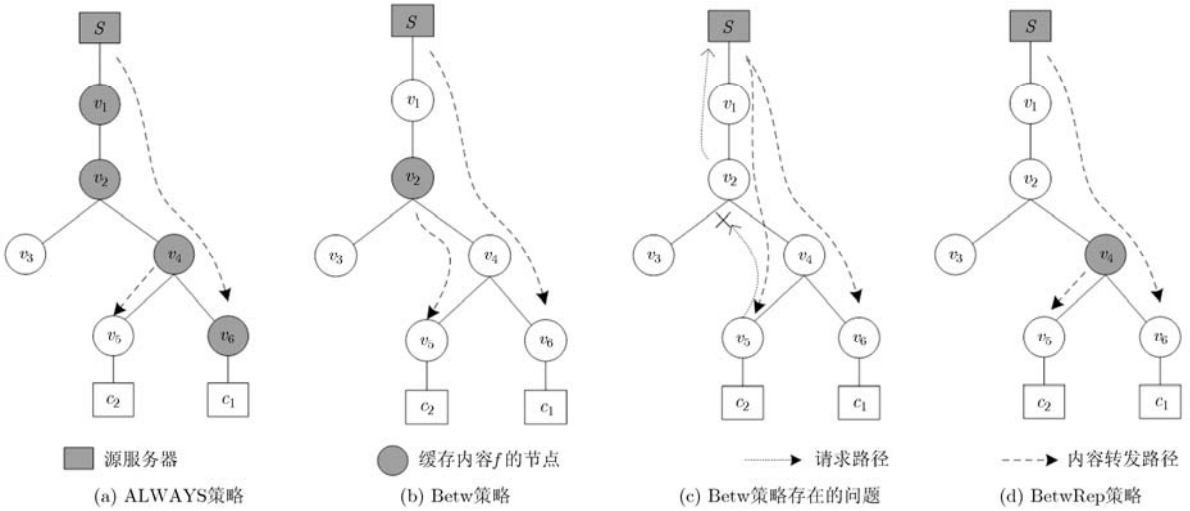


图 1 ALWAYS, Betw 和 BetwRep 缓存策略实例

来决定路径上的缓存节点，兴趣包请求路径 L 上具有最大度量的节点 v_k 缓存返回内容，而其它节点不再缓存，由此得到一个基于节点介数和缓存替换率作为缓存决策度量的新型网内缓存策略 BetwRep。根据 BetwRep 缓存策略，如图 1(d)所示，在由 c_1 发出的兴趣包获得内容所经过的路径上，节点 v_2 的介数最大，v_4 次之，但如果 v_2 的缓存替换率远大于 v_4，则通过比较新的度量 $M(v)$，使得返回内容被缓存在节点 v_4，由此来自 v_5 的请求，会直接在 v_4 获得内容，从而提高 CCN 网络网内缓存性能。

3.2 BetwRep 缓存策略的实现

由于节点的介数度量和内容缓存更替度量无法直接关联，因此 BetwRep 缓存策略中为计算节点度量 $M(v)$，需要对这两个度量进行归一化。节点介数 $\mathrm{Betw}(v)$ 是一个相对值，是指某个节点的介数与本请求路径 L 上所有节点最大介数的比值，表征的是节点在本路径 L 上的重要程度，其中

$$\mathrm{Betw}(v)=\frac{B(v)}{\max\limits_{v,l\in L}\{B(l)\}},\ 0\le \mathrm{Betw}(v)\le 1 \tag{2}$$

$$B(v)=\sum_{s\ne v\ne t\in V}\frac{\sigma_{s,t}(v)}{\sigma_{s,t}} \tag{3}$$

下面引入节点的缓存更替率，用 $\mathrm{Re}(v)$ 表示。

$$\mathrm{Re}(v)=\frac{\sum\limits_{i=1}^{m}S(f_i)}{C(v)} \tag{4}$$

其中 $S(f_i)$ 是节点 v 被替换内容 f_i 的大小(单位是 chunk)，$C(v)$ 是节点 v 的缓存大小，m 是单位时间内节点 v 被替换的内容个数。归一化后，

$$\mathrm{Repl}(v)=\frac{\mathrm{Re}(v)}{\max\limits_{v,l\in L}\{\mathrm{Re}(l)\}},\ 0\le \mathrm{Repl}(v)\le 1 \tag{5}$$

设计新的节点度量 $M(v)$，定义如式(6)。

$$M(v)=\frac{\mathrm{Betw}(v)}{\mathrm{Repl}(v)},\ \mathrm{Repl}(v)\ne 0 \tag{6}$$

$$v_k=\arg\max_{v\in L}\{M(v)\} \tag{7}$$

其中 $\mathrm{Betw}(v)$ 是节点介数的相对度量，$\mathrm{Repl}(v)$ 是缓存更替率的相对度量，若 $\mathrm{Repl}(v)=0$，表示节点缓存未满或者没有新内容到达，为了使节点度量表达式统一，此时取 $\mathrm{Repl}(v)=\varepsilon$，$\varepsilon$ 是一个非常小的正数。

BetwRep 缓存策略是基于单个兴趣包请求的，本文将结合节点对兴趣包和数据包的处理流程，给出 BetwRep 缓存策略的具体实现。当兴趣包在节点未能取得缓存命中且没有请求内容的 PIT(Pending Interest Table)条目时，都要记录经过节点的介数、缓存替换率和跳数的 3 元组$(B(v),\mathrm{Re}(v),H)$，并一同转发到下个节点。

兴趣包在节点缓存命中，则取出 3 元组集合做上述运算，找出兴趣包经过路径上度量 $M(v)$ 最大的那个节点 v_k，设距离 $D=H_{\max}-H_k$，$H_{\max}$ 是请求的最大跳数，H_k 是请求到节点 v_k 的跳数，将$(B(v_k),D)$添加到数据包头里，数据包每到达一个节点，比较介数与 $B(v_k)$ 的关系，若两者相等，且距离 D 为零，数据包就在该节点缓存，否则 D 减 1，被转发到下一个节点。BetwRep 策略只跟节点度量有关，所以兴趣包缓存命中节点可以独立作出判决，节点间不需要进行额外的信息交互，而且节点的度量介数是拓扑形成后就预计算好的，表征着节点在拓扑中的重要程度，缓存内容替换率是描述节点缓存繁忙程度的，并通过实时统计获得，然后添加到经过的兴趣包中。

如果兴趣包在节点上没有实现内容缓存命中，但已有请求内容的 PIT 条目，此时，需要将兴趣包的信息和到达接口(face)记录到 PIT 表中，并丢弃兴趣包。根据所提出的 BetwRep 策略，在丢弃兴趣包之前，需要像缓存命中那样对兴趣包所携带的节点参数提取并进行运算，选择出从对应接口转发的内容在兴趣包经过路径上应缓存的节点。为了便于实现，将为被聚合的兴趣包对应接口设定一个标签，并计算出要缓存的节点，将该节点的缓存标签$(B(v_k),D)$添加到接口记录中，当请求内容返回节点时，将接口设定的标签替换掉内容包中的缓存标签，然后再从接口转发。为了更清晰地说明 BetwRep 缓存策略实现过程，表 1 给出了节点兴趣包和内容数据包处理的伪代码(Pseudocode I 和 Pseudocode II)。

4 仿真分析

为验证本文提出的 BetwRep 缓存策略对于 CCN 网络性能的提升，本文通过 ndnSIM[15,16]仿真器实现了对 Betw, ALWAYS 和 BetwRep 3 种缓存策略的仿真，并对它们的性能进行了定量的比较和分析，仿真结果验证了 BetwRep 缓存策略的有效性。

4.1 仿真环境和参数配置

本实验硬件环境为Intel(R)Core(TM) i3-2120 CPU@3.30 GHz, 4 GB 内存。操作系统是Ubuntu 12.04 LTS 64 bit。仿真环境ndnSIM仿真器是基于NS3的一个NDN仿真模块，该模块实现了对NDN基本功能的模拟。我们通过对其进行代码修改，实现本文提出的缓存策略。将仿真数据导入Matlab软件中处理，仿真结果如图2和图3所示。主要参数如下：

表 1 节点兴趣包伪代码

Pseudocode I 兴趣包处理伪代码:

Initialize (serials $B(v)$=[0,⋯,0],Re(v)=[0,⋯,0]))
Initialize (serials Betw(v)=[0,⋯,0],Repl(v)=[0,⋯,0])
for each (v_k from i to j)
If data in cache || entry in PIT
 Get $B(v)$, Re(v)
  Calculate Betw(v), Repl(v)
   Calculate
    $M(v)$ = Betw(v) / Repl(v))
    v_k = arg max$\{M(v)\}$
   B=$B(v_k)$
    If data in cache
    then send(data)
    else create ($\le B_k, D_k \ge$)
        discard interest_packet
  else
    Get $B(v_k)$, Re(v_k)
  $B(v)_k$ =$B(v_k)$, Re$(v)_k$=Re(v_k)
  forward request to the next hop towards j

Pseudocode II 数据包处理伪代码:

Record Betw from corresponding content request
  for each (v_k from j to i)
  Get $B(v_k)$, $\le B, D \ge$
  if B == $B(v_k)$ && 0==D
  then cache(data)
  else if $\le B_k, D_k \ge$!=NULL in face
        $\le B, D \ge$ =$\le B_k, D_k \ge$
        send(data)
  else
  forward data packet to the next hop toward

总内容数K=10000个文件，单个文件大小设为一个chunk，内容请求服从Zipf-Mandelbrot分布，参数初始化值分别是：漂移量$q=0.7$，幂律指数$s=0.7$，无符号随机数SeqRng $(0.0, 1.0)$ [15,17]；节点请求到达服从泊松分布，λ=100 个/s；假设每个节点缓存容量相同为C，为对比系统性能指标随节点缓存大小的变化，仿真中C分别取100, 200, 400, 600,⋯,2000，单位为块(chunk)；则式(5)变成

$$\mathrm{Repl}(v)=\frac{k_v}{\max\limits_{v,l\in L}\{k_l\}},\quad 0\le \mathrm{Repl}(v)\le 1 \tag{8}$$

请求转发方式选择洪泛模式，节点缓存替换策略选择 LRU。根据 CCN 的通信机制，节点具有兴趣包的聚合特性，返回内容沿着组播树分发，因此仿真网络选择树状拓扑结构，本文选取一个深度为 7，总节点数为 17 的随机树状拓扑。

4.2 仿真结果分析

考虑两个关键的性能指标：请求平均跳数$\eta(t)$和内容源端命中率$\gamma(t)$。其中

$$\eta(t)=\left(\sum_{r=1}^{Q}h_r(t)\right)\Big/Q \tag{9}$$

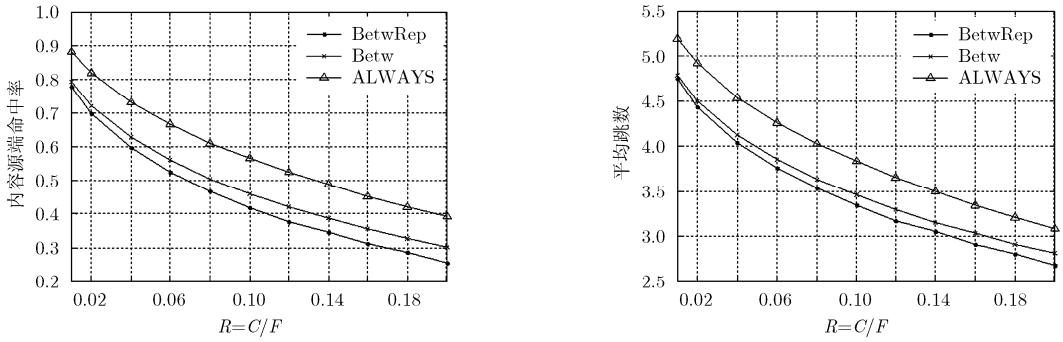
$$\gamma(t)=\left(\sum_{r=1}^{Q}w_r(t)\right)\Big/Q \tag{10}$$

$$w_r(t)=\begin{cases}1, & \text{请求}r\text{在内容源端命中}\\ 0, & \text{其它}\end{cases} \tag{11}$$

这里，$h_r(t)$表示时间$t-1$到t时间内的请求r获得内容所经过的跳数，$w_r(t)$是个二值函数，值为 1 时表示请求r在内容源端命中，反之取 0，Q是总请求数。内容源端命中率越低，说明请求的网内命中率越高，大量的请求可以在靠近请求端获取内容，减轻了内容源端的负载，与之相应的是内容传输跳数的减少，带来带宽消耗的降低。但是平均跳数的减少与内容源端命中率的降低并不是直接成比例的，例如一个请求在远离和靠近内容源端的两个节点缓存命中所经过的跳数是不同的，因此本文把平均跳数$\eta(t)$和内容源端命中率$\gamma(t)$作为两个独立的性能指标来考察。

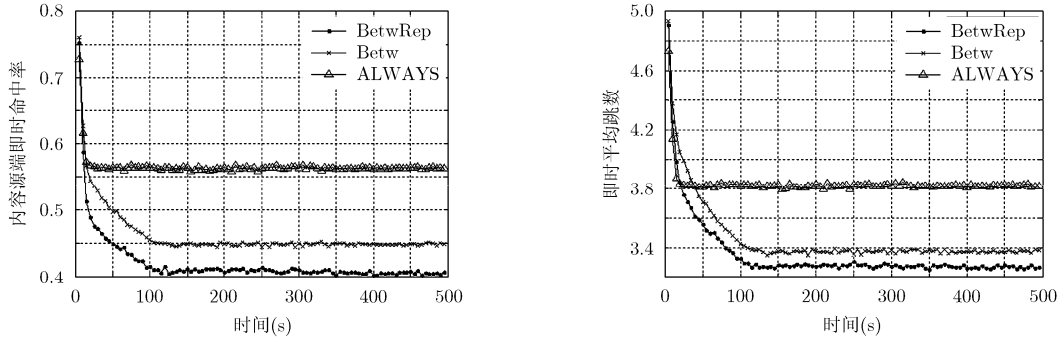
图 2 是 ALWAYS, Betw 和 BetwRep 3 种缓存策略在节点缓存容量变化情况下，请求的平均跳数和内容源端命中率的仿真对比。为描述方便，把缓存大小与内容总量之比标记为R，取值范围在 0.01 ~0.2 之间，即 C 的取值范围在 100~2000 chunks 之间，R=C/F，C 是节点的容量，F 是内容总量，两者单位是块(chunk)，作出 BetwRep 缓存策略性能指标与R的关系图 2。由图可知，在节点缓存大小变化过程中，BetwRep 策略都取得了比较稳定的性能提高，相比 Betw 策略内容源端命中率平均下降近 5%，平均跳数减少 0.2 跳。相比 ALWAYS 策略性能提高更加明显，内容源端命中率平均下降近 15%，平均跳数减少 0.5 跳。

图 3 是节点缓存大小和内容总量之比R取定后，ALWAYS, Betw 和 BetwRep 3 种缓存策略仿真时间内瞬时性能的比较。类似文献[9]，设$R=0.1$。从收敛速度看，ALWAYS 策略具有一定的优势，Betw 和 BetwRep 收敛速度相当。但达到稳定状态后，BetwRep 缓存策略的性能优势较为明显，相比 ALWAYS 策略，内容源端命中率平均下降 17%。相对于 Betw 策略，性能优势依然明显。由于仿真是



(a)内容源端命中率随节点缓存大小的变化 (b)平均跳数随节点缓存大小的变化

图 2 网络缓存性能与节点缓存大小关系图



(a)内容源端即时命中率 (b)即时平均跳数

图 3 网络即时缓存性能图

从 $t=0$ 开始的，而数据统计数据是从 $t=0.1$ s 开始，所以图 3(a)内容源端即时命中率初始值并没有达到 1，但命中率出现了急剧下降的走势，这也正是 CCN 网络给内容分发带来收益的体现。

5 结束语

为了解决内容中心网络中返回内容在请求路径上如何选择缓存节点的问题，本文将节点的内容缓存替换率和节点介数作为参数构造出一个新的度量，以此提出了一个新的 BetwRep 缓存策略。该策略避免了 ALWAYS 策略带来的大量缓存冗余，同时克服了 Betw 缓存策略中节点选取只考虑节点的重要性而导致重要节点内容替换频繁，流行内容在节点中缓存时间变短的缺点。通过对 3 种方案两个性能指标平均跳数和内容源端命中率的对比发现，本文提出的 BetwRep 缓存策略，性能比 Betw 策略有了 5%的提高，相比 ALWAYS 策略性能提高更加显著。在后续工作中，我们将在本文基础上通过引入内容的流行度，设计出基于概率的缓存策略。

参 考 文 献

[1] Jacobson V, Smetters D K, Thornton J D, *et al*.. Networking named content[C]. Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, Rome, December 1–4, 2009: 1–12.

[2] Yi C, Afanasyev A, Wang L, *et al*.. Adaptive forwarding in named data networking[J]. *ACM SIGCOMM Computer Communication Review*, 2012, 42(3): 62–67.

[3] Kideok Cho, Lee Mun-young, Park Kun-woo, *et al*.. WAVE: popularity-based and collaborative in-network caching for content-oriented networks[C]. IEEE Conference on Computer Communications (INFOCOM WKSHPS), Workshops, Orlando, March 25–30, 2012: 316–321.

[4] Dan A and Towsley D. An approximate analysis of the lru and fifo buffer replacement schemes[C]. Proceedings of the 1990 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, Boulder, May 22–25, 1990: 143–152.

[5] Leet Dong-hee, Choit Jong-moo, Kim Jong-hun *et al*.. On the existence of a spectrum of policies that subsumes the Least Rrecently Used (LRU) and Least Frequently Used (LFU) policies[C]. Proceedings of the 1999 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, Atlanta, May 1–4, 1999: 134–143.

[6] Jelenkovic P, Radovanovic A, and Squillante M S. Critical sizing of lru caches with dependent requests[J]. *Journal of Applied Probability*, 2006, 43(4): 1013–1027.

[7] Lee Breslau, Pei Cao, Li Fan *et al.*. Web caching and Zipf-like distributions: evidence and implications[C]. Proceedings of INFOCOM'99, Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, NewYork, March 21–25, 1999: 126–134.

[8] Laoutaris N, Syntila S, and Stavrakakis I. Meta algorithms for hierarchical web caches[C]. Proceedings of the IEEE International Performance Computing and Communications Conference (IEEE IPCCC), Phoenix, April 15–17, 2004: 445–452.

[9] Chai Wei-koong, He Di-liang, Ioannis P, *et al.*. Cache "Less for More" in information-centric networks[C]. Proceedings of the 11th International IFIP TC 6 Conference on Networking, Prague, May 21–25, 2012: 27–40.

[10] Chai Wei-koong, He Di-liang, Ioannis P, *et al.*. Cache "Less for More" in information-centric networks (extended version)[J]. *Computer Communications*, 2013, 36(7): 758–770.

[11] Ghodsi A, Koponen T, Raghavan B, *et al.*. Information-centric networking: seeing the forest for the trees[C]. Proceedings of the 10th ACM Workshop on Hot Topics in Networks, Cambridge, Massachusetts, November 14–15, 2011: 1–6.

[12] Goh K, Oh E, Kahng B, *et al.*. Betweenness centrality correlation in social networks[J]. *Physical Review E*, 2003, 67(1): 1–4.

[13] Izquierdo L R and Hanneman R A. Introduction to the formal analysis of social networks using mathematics[OL]. http://www.luis.izquierdo.name, 2006.

[14] Rossi D and Rossini G. Caching performance of content centric networks under multi-path routing (and more)[R]. Technical Report, Telecom ParisTech, 2011.

[15] Afanasyev A, Moiseenko I, and Zhang L. ndnSIM: NDN simulator for NS-3[R]. Technical Report, NDN-0005, NDN Project (July 2012).

[16] University of California, Los Angeles. NS-3 based Named Data Networking (NDN) simulator[OL]. http://ndnsim.net/, 2012.

[17] Tsinghua University, P. R. China. ndn-consumer-zipf-mandelbrot[OL]. http://ndnsim.net/doxygen/ ndn-consumer-zipf-mandelbrot_8cc_source.html, 2012.

崔现东： 男，1980 年生，博士生，研究方向为未来网络关键技术、内容中心网络.

刘　江： 男，1983年生，讲师，研究方向为网络虚拟化、内容中心网络.

黄　韬： 男，1980年生，副教授，研究方向为路由与交换、内容中心网络.

陈建亚： 男，1951 年生，教授，研究方向为下一代网络、路由与交换.

刘韵洁： 男，1943 年生，中国工程院院士，博士生导师，主要研究领域为未来网络、网络融合等.