

# 基于概率的启发式 ICN 缓存内容放置方法

吴海博<sup>1</sup>, 李俊<sup>1</sup>, 智江<sup>1,2</sup>

(1. 中国科学院计算机网络信息中心, 北京 100190; 2. 中国科学院研究生院, 北京 100049)

**摘要:** 提出一种基于概率存储的启发式住处中心网络内容缓存方法 (PCP)。主要思想是请求消息和数据消息在传输过程中统计必要信息, 当数据消息返回时, 沿途各缓存节点按照一定概率决策是否在本本地缓存该内容。设计缓存概率时综合考虑内容热度和缓存放置收益, 即内容热度越高, 放置收益越大的内容被缓存的概率越高。实验结果表明, PCP 在缓存服务率、缓存命中率、平均访问延迟率等方面, 与现有方法相比具有显著优势, 同时 PCP 开销较小。

**关键词:** 信息中心网络; 网内缓存; 内容放置; 协作缓存

**中图分类号:** TP393

**文献标识码:** A

## Probability-based heuristic content placement method for ICN caching

WU Hai-bo<sup>1</sup>, LI Jun<sup>1</sup>, ZHI Jiang<sup>1,2</sup>

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

2. Graduate School, Chinese Academy of Sciences, Beijing 100049, China)

**Abstract:** A probability-based heuristic caching strategy (PCP) was proposed. The major idea was that relevant information was collected when the interest and data packet are transmitted. When the data packet is returned back, on-path cache nodes decide whether to cache the content with certain probability. The probability is associated with the content popularity and placement benefit, i.e., proportional to content popularity and content placement benefit. The experimental results indicate PCP can achieve salient performance gain in terms of cache service ratio, cache hit ratio, access latency ratio, compared with current strategies. Meanwhile, PCP's overhead is affordable.

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

### 1 引言

互联网最初的设计目标是实现资源共享, 主要采用主机间端到端 (host-to-host) 通信模式。然而经过四、五十年的发展, 目前以视频分发、文件下载为代表的信息获取类服务已成为当前互联网的主要应用模式<sup>[1]</sup>。当前网络用户在获取网络信息时, 更关注信息本身, 而并不关心信息的位置和来源, 然而, 目前互联网信息传输依然沿用传统的基于 IP

地址的端到端通信模式, 因此, 传统网络通信模式已不再适用于新兴的应用需求。在此背景下, 以 CCN/NDN<sup>[2]</sup>、DONA<sup>[3]</sup>、NetInf<sup>[4]</sup>、PURSUIT<sup>[5]</sup>等为典型代表的信息中心网络 (ICN, information centric networking) 应运而生, 并得到广泛关注和研究。ICN 以信息为中心, 用户只需关注信息 (内容) 本身, 而不必关注信息的来源, 从而使网络通信模式与应用需求相适应。

网内缓存 (in-network caching) 是 ICN 的重要

收稿日期: 2015-08-12; 修回日期: 2015-11-30

通信作者: 李俊, lijun@cnic.cn

基金项目: CNIC “一三五” 所级专项基金资助项目 (No.CNIC\_PY-1401); 国家重点基础研究发展计划 (“973” 计划) 基金资助项目 (No.2012CB315803); 中国科学院知识创新工程青年人才领域基金资助项目 (No.CNIC\_QN\_1508); 中国科学院重大装备研制基金资助项目

**Foundation Items:** Around Five Top Priorities of “One-Three-Five” Strategic Planning, CNIC (No.CNIC\_PY-1401), The National Basic Research Program of China (973 Program) (No.2012CB315803), The Knowledge Innovation Program of the Chinese Academy of Sciences (No.CNIC\_QN\_1508), The Significant Equipment Development Program of Chinese Academy of Sciences

特征,其核心思想是通过网络内部路由器对内容进行缓存,此后用户请求在网络中路由,沿途若命中缓存则可直接响应,而不必每次从内容源(服务器)获取,通过就近服务的方法,可达到提高内容传输效率、降低网络流量以及缓解服务器访问压力等目的。

缓存系统设计的优劣,特别是内容放置的合理性,对 ICN 内容分发效率具有重要影响。合理的缓存内容放置策略会使用户更易于从临近缓存节点中获取内容,而无需每次从内容服务器获取,从而有效降低访问延迟、网络流量以及服务器负载。然而,当前 ICN 缓存仍面临以下 3 方面的突出问题。

1) 空间利用率低。一方面,缓存内容放置不合理,不同缓存间存在大量重复数据,内容缓存冗余度较高,缓存间几乎缓存相似的内容,空间利用率较低。另一方面,没有充分考虑缓存内容访问频率的差异,大量冷门内容占用众多宝贵的缓存资源,但无法带来收益。

2) 缓存命中率低。缓存命中率不高,致使请求消息大量“扑空”,每次仍需从较远的缓存节点或内容服务器获取内容,然而途径缓存节点需查询、维护等操作,引入大量延迟开销,影响内容传输效率。

3) 访问延迟较大。内容缓存的其中一个重要目的是为了降低内容访问延迟,改善用户体验。现有研究大多致力于提高缓存空间利用率,改善缓存命中率,然而,命中率的提升未必能够降低内容访问延迟。

为了解决上述缓存空间利用率低、缓存命中率低、访问延迟长等问题,本文提出一种启发式的基于概率的内容缓存方法(PCP, probability-based content placement)。本文的主要贡献如下。

1) 对内容缓存问题进行建模研究,分析发现内容热度和放置收益是影响缓存收益的两大因素。

2) 提出了一种基于概率的 ICN 缓存内容放置方法,综合考虑内容热度和内容放置收益,有效提高缓存空间利用率以及缓存服务效率。

3) 定义多种 ICN 缓存性能评价指标,同时与现有 ICN 缓存代表性方案对比,并在多种场景下开展实验研究,验证 PCP 的性能优势。

## 2 相关工作

按照缓存存放位置与内容源与请求者间路径的相对关系,可将 ICN 缓存现有研究分为“路径之上缓存”(on-path caching)和“路径之外缓存”(off-path

caching)。如果缓存存放在内容源与请求者间的路径上,则为路径之上缓存,反之,则称为路径之外缓存。

### 2.1 路径之上缓存

路径之上缓存策略将内容潜在地存储在请求者与内容源间的路径上,代表方案有以下几种。

CCN/NDN 早期的方案是采用随处缓存的方案(LCE, leave copy everywhere)<sup>[2]</sup>,内容在返回途中被沿途所有缓存节点缓存。该方案会导致大量内容重复存储的情况,空间利用率较低,难以发挥内容缓存的性能优势。

Laoutaris 等<sup>[6]</sup>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,仅在当前节点的第一个下游节点中缓存此内容,从而有效降低了缓存内容的冗余度。作者同时提出另一种 MCD(move copy down)缓存方案,与 LCD 相比的区别在于,在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。

Psaras 等<sup>[7]</sup>设计了一种基于概率的 ProbCache 缓存机制,同时兼顾缓存容量以及缓存收益,缓存节点离服务器越远(即缓存收益越大),内容被缓存在该节点中的概率越高;同时沿途下游节点总缓存容量越大,在当前节点中被缓存的概率越大。然而,该方法没有区分不同内容请求的差异,特别是没有考虑内容的不同热度,致使任何内容在同一缓存节点被缓存的概率是相同的。

Cho 等<sup>[8]</sup>提出一种 WAVE 缓存机制,上游节点以文件为单位记录内容被请求的次数,同时向下游节点建议进行缓存的块数,该数值按照请求次数的多少,以指数级速度增长。该方法能使热门内容以指数级的速度快速扩散,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。

Huo 等<sup>[9]</sup>提出一种基于相关性概率的 ICN 协作缓存策略,根据路径及相邻节点信息做出本节点的缓存判断,从而有效控制缓存冗余。

### 2.2 路径之外缓存

在路径之外缓存策略中,内容可以存储在路径之外的其他位置,目标是改善内容的可用性,通常借助缓存间协作机制实现。

Rosensweig 等<sup>[10]</sup>提出一种隐式协作的 Breadcrumbs 路由机制。数据返回时,不仅沿途缓存节点缓存内容,同时在返回路径上保留轨迹信息。轨迹信息五元组包括:内容全局唯一标识、内容到达的上游节点编号、内容到达的下游节点编号、内容最近经过该节点的时刻、节点收到的内容最近被请求时刻。当请求消息命中缓存时,直接返回相关内容。如果缓存未命中,则按照轨迹信息决定是向上游源节点转发,还是沿之前分发路径向下游转发。如果按照轨迹沿下游转发失败,请求消息仍会向内容源转发。然而,该方案需要记录每个内容的相关统计信息,开销较大,且有可能存在路由环路的情况。

Zhu 等<sup>[11]</sup>提出一种端系统间相互协作的 CONIC 机制。缓存内容存放在用户端,路由器中维护相关索引信息。请求消息在路由节点的引导下,在邻近用户节点中快速命中缓存,以提高缓存的服务效率。该方法易部署,同时具有类似于 P2P 系统的良好可扩展性。然而,该方案中路由器仅用于存储内容索引,不缓存具体内容,因此难以充分发挥网内缓存的优势。

Li 等<sup>[12]</sup>提出了一种基于散列函数的邻居节点间协作的 CINC 机制。当缓存节点收到某内容块(chunk)时,根据块的编号结合散列函数,可以计算出存放该数据块的节点编号;同理,当请求到达时,可以引导到就近的缓存节点中。该方案中节点标号需要合理的设计,以使协作节点间距离之和最小,从而充分发挥缓存的整体性能;同时,当发生节点失效时,系统的恢复开销较高。

Ming 等<sup>[13]</sup>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。

Wang 等<sup>[14]</sup>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 的性能,然而两者相差不大;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<sup>[15]</sup>提出了一种同时考虑内容的放置、发现和替换 APDR 机制。请求消息沿途统计各缓存节点的潜在需求、空闲缓存等信息,请求消息汇聚点以及目的节点据此计算缓存方案,并将该方案通过数据消息返回,在沿途指定缓存节点以及缓存承诺时间。然而,该方法中请求消息需要携带沿途信息,缓存决策过程计算复杂,这些均会引入一定的维护和计算开销。

### 3 ICN 概述

针对 ICN 未来互联网体系结构代表性的研究主要有 CCN/NDN<sup>[2]</sup>、PURSUIT<sup>[5]</sup>、DONA<sup>[3]</sup>、SAIL<sup>[4]</sup>等。它们虽然体系结构间存在差异,然而均以网内节点缓存内容为共同特征。同时,请求消息路由和内容访问过程大致相同:请求消息(interest packet)中携带内容名字,路由节点基于该名字进行逐跳转发。当请求消息到达内容源或缓存时,数据消息(data packet)采用反向路径的方式返回给用户。

由于 NDN<sup>[2]</sup>是 ICN 的典型代表,因此本文的机制主要针对 NDN 场景进行介绍,同时易于扩展到其他 ICN 系统。在 NDN 系统中,每个路由器节点上需要维护 3 张表:CS(content store)用来缓存数据并且实时更新缓存内容;FIB(forwarding information base)由基于名字标识的路由协议生成,用于指导请求消息的转发;PIT(pending interest table)用于记录请求消息的到达接口,从而建立返回数据分组的转发路径,同时具有聚合相同内容请求的作用。路由节点在收到数据请求分组后,首先查询 CS 表,如果命中则返回数据,若没有命中则查询 PIT 表,PIT 表中如果没有该请求消息的记录则加入 PIT 表,然后根据 FIB 表选择一个或多个接口转发该请求消息。如果 PIT 表中已有相同内容的请求,则将当前请求加入到 FIB 表中,同时不再转发该请求消息。当返回的数据分组按照由 PIT 表建立的反向路径传输时,各个路由节点会根据一定的策略对数据进行缓存,然后再按照 PIT 表中记录的请求消息来源信息,将数据复制多份并转发给各个相应接口。

### 4 问题建模

#### 4.1 系统模型

由于 NDN 是 ICN 的典型实例,因此本文以

NDN 为例来阐述所提出的策略。用户请求消息基于 FIB 表进行逐跳转发,各缓存节点(路由器)基于 OSPF<sup>[16]</sup>或 NLSR<sup>[17]</sup>等域内路由协议,相互间交换路由信息,并最终生成 FIB 表。相应的数据消息基于 PIT 表中记录的状态信息,沿反向路径传回到请求者。所有内容以数据块(chunk)为单位,并且所有块具有相同大小<sup>[18]</sup>。

#### 4.2 内容放置最优化模型

内容放置问题定义为在各缓存节点部署容量(反映成本)一定的前提下,如何将所有内容块合理分配到各缓存节点,以最大化缓存收益。相关符号定义如表 1 所示。

表 1 符号说明

符号	说明
$\lambda_i$	节点 $v_i$ 收到的关于内容块 $c$ 的平均请求速率
$p_c$	内容块 $c$ 的热度(被请求的概率)
$b_{c,i,r}$	将内容块 $c$ 缓存于节点 $v_i$ 时对于请求者 $v_r$ 而言获得的收益
$e_{c,i}$	二元变量,表示是否在节点 $v_i$ 处缓存内容块 $c$
$V$	所有缓存节点的集合
$C$	所有内容块的总容量(以内容块为单位)
$C(i)$	缓存节点 $v_i$ 的最大容量(以内容块为单位)

针对 ICN 缓存内容放置问题,可构造如下优化模型

$$\text{Max} \quad \sum_{c \in C} \sum_{i \in V} \sum_{r \in V} \lambda_i p_c b_{c,i,r} e_{c,i} \quad (1)$$

$$\text{s.t.} \quad \sum_{c \in C} p_c = 1 \quad (2)$$

$$e_{c,i} \in \{0/1\} \quad (3)$$

$$\sum_{c \in C} e_{c,i} \leq C(i) \quad (4)$$

其中, $\lambda_i$ 表示缓存节点  $v_i$  接收到的所有用户请求的平均速率; $p_c$ 是内容块  $c$  的热度,代表内容块  $c$  被访问的频率(已被归一化); $b_{c,i,r}$ 表示将缓存节点  $v_r$  相邻用户请求的内容块  $c$  时,如果将  $c$  放置在缓存节点  $v_i$ ,相比于从服务器获取所获得的跳数收益; $e_{c,i}$ 是一个二元变量,表示是否将内容块  $c$  放在缓存节点  $v_i$ 。式(4)表示每个缓存节点所放置的内容,需满足其自身容量限制。

#### 4.3 问题分析

由于目标函数与约束条件均为线性,且自变量  $e_{c,i}$  的取值为整数,因此该问题是整数线性规划(ILP, integer linear programming)问题,是 NP 完全问题,

难于直接求解。NP 完全问题的简略证明如下。首先,该问题任意一个可能的解,都可以在多项式时间内判断其是否是可行解,因此该问题是 NP 问题。其次,通过将该 ILP 问题归约为 3-CNF-SAT 问题,以证明其是 NP 完全问题。

同时,即使可以直接求解该问题,得到的是一种全局集中式的缓存内容放置方案,一方面,计算需要全局拓扑结构、内容热度和全局路由等信息,往往难以获得;另一方面,由于缺少集中控制节点,难以在分布式的缓存系统中实现。因此,本文提出一种分布式缓存内容放置算法,作为一种实际可行的方案,所有缓存设备基于局部信息独立进行缓存决策。

如果假设所有用户均具有相同的请求速率,因此变量  $\lambda_{c,i}$  的影响可以被忽略。当决策变量  $e_{c,i}=1$  时,为最大化目标函数,应最大化内容热度  $p_c$  和内容放置收益  $b_{c,i,r}$  的乘积。因此,可以看出内容热度和内容放置收益是影响缓存收益的 2 个关键因素。基于 2 个关键因素,本文提出一种启发式内容放置算法,缓存节点可以采用分布式的内容缓存策略,以优化缓存系统的整体性能。

### 5 PCP 方案

假定 ICN 网络由内容服务器、具有缓存功能的路由器节点以及一些用户组成。ICN 网络中存在请求消息和数据消息 2 种分组。只有内容服务器会产生并永久保存内容。路由节点可以缓存途径的内容,同时维护和查找 CS、PIT、FIB 这 3 张表。内容以分块方式进行缓存和传输,且所有内容块的大小相同。用户可以通过发送请求消息的方式,从 ICN 网络中的内容服务器或缓存节点获取相应内容。

#### 5.1 PCP 核心思想概述

方案总体思路是每个缓存节点对所有经过的数据块,按照一定概率决策是否对其缓存,且缓存概率与内容热度和缓存收益成正比。内容热度越高、缓存收益越大的内容块,其被缓存的概率就越高。其中,缓存收益定义为从缓存节点获取内容相比于从服务器获取内容所获得的跳数的节省。

**定义 1** 缓存收益。若请求者  $u$  从缓存节点  $i$  中获取内容块  $c$  所需跳数为  $h_c(u,i,c)$ ,从服务器  $s$  中获取内容块  $c$  所需跳数为  $h_s(u,i,c)$ ,则缓存收

益  $b(u,i,c)$  被定义为所节省的跳, 即  $b(u,i,c) = h_s(u,i,c) - h_c(u,i,c)$ 。

定义 2 相对缓存收益  $r(u,i,c)$  为缓存收益  $b(u,i,c)$  与请求者与服务器之间的跳数  $d(u,s)$  之比,  $r(u,i,c) = \frac{b(u,i,c)}{d(u,s)}$ 。

定义 3 内容热度。内容热度表示内容在全网范围内被访问的次数, 代表内容的全局热度。

定义 4 内容相对热度( $p_c$ )为内容热度与最热门内容块的热度比值。

## 5.2 缓存概率公式

缓存节点  $i$  缓存内容  $c$  的概率可表示为

$$P(c,i) = p_c r(u,i,c) \quad (5)$$

其中,  $p_c$  代表内容块  $c$  的相对热度,  $r(u,i,c)$  表示缓存的相对收益, 二者均为分数, 便于归一化到 0~1 之间。

## 5.3 相关信息统计

公式中涉及到内容热度、缓存收益以及请求者与服务器间的距离 3 个量, 本节主要介绍三者的获取方法。缓存收益和路径长度需要缓存节点间相互协作来获取, 围绕请求消息和数据消息的处理, 本文提出几个相关算法, 具体内容参见 5.4 节。

### 5.3.1 内容热度的统计

内容热度可以基于历史统计信息获得。例如, 网络中各路由器节点实时统计本地内容热度, 此后通过相互间通信, 进而获得整个网络的内容全局热度。本文在此不将热度的统计方法作为研究重点, 并假设内容热度已通过某种方式获得。

### 5.3.2 收益和距离的统计

为完成缓存收益以及请求者与服务器间的距离的统计, 需对 ICN 中消息格式和表结构进行重新定义。

#### 1) 消息格式定义

对于请求消息, 除传统的 Content Name、Selector、Nonce 字段外, 新增 Hop 字段, 用于记录从用户到达服务器或命中缓存时所经过的跳数。

对于数据消息, 除传统的 Content Name、Signature、Signed Info、Data 字段外, 新增 2 个字段: 收益 (Benefit) 字段, 用于记录从当前节点提供缓存内容相比于从内容服务器获取内容所获得的收益; 距离 (Distance) 字段, 用

于记录用户到内容源所经过的跳数, 当请求消息到达内容服务器时, 由请求消息中 Hop 字段赋值获得。

#### 2) 表结构定义

对 CS 表和 PIT 表作如下扩展, FIB 表的格式保持不变。每个缓存节点查表操作的顺序依然为 CS-PIT-FIB。

对于 CS 表, 除传统 Name、Data 字段外, 新增 Benefit 字段, 用于记录相关内容从本地缓存获取相比于从服务器获取的收益, 此处主要指跳数节省。其目的在于, 当请求在缓存命中时, 借助请求中 Hop 字段和 Benefit 字段, 仍能统计从用户到内容服务器间的距离。

对于 PIT 表, 除传统的 Prefix 和 Requesting Faces 字段外, 新增 Hop 字段, 用于记录此前请求消息从用户到达此节点的跳数。其目的在于, 当请求在缓存命中时, 借助数据消息中的 Benefit 字段和 PIT 表记录的 Hop 值, 将两者相加, 便于统计各接口发送请求的用户到内容服务器间的距离。

#### 3) 具体统计方法

具体信息统计方法在 5.4 节中进行介绍。

## 5.4 具体算法

按照请求消息和数据消息的处理方法不同, 本节介绍 3 个算法。算法 1 用以描述服务器对请求消息的处理流程; 算法 2 用以描述缓存节点对请求消息的处理流程; 算法 3 主要描述缓存节点对数据消息的处理流程。

### 5.4.1 Benefit 数值的统计

Benefit 数值记录在数据消息的 Benefit 字段中, 表示如果当前缓存节点提供内容服务, 相比于服务器提供内容服务所获得的跳数收益。对其有如下几种情况的处理。

#### 1) 请求消息到达服务器

当请求消息到达内容服务器时, 数据消息中的 Benefit 字段初始化为 1。此后, 在数据消息返回的过程中, Benefit 字段逐跳加 1, 因此沿途缓存节点均能从数据消息中获得。如算法 1 所示。

#### 2) 请求消息命中缓存

数据消息中的 Benefit 字段可直接从 CS 表中 Benefit 字段读出此值。此后, 在数据消息返回的过程中, Benefit 字段逐跳加 1 (算法 3 中第 6) 行), 因此沿途缓存节点均能从数据消息中获得。

### 3) 数据消息到达 PIT 表

按照 PIT 的表项依次根据相应接口向下转发数据消息, 此时各接口的数据消息中的 Benefit 值与原数据消息中的 Benefit 值相同。如算法 3 第 5)~9) 行所示。

### 5.4.2 Distance 数值的统计

Distance 数值记录在数据消息的 Distance 字段中, 表示作为请求者的用户与内容服务器间的跳数距离。

#### 1) 请求消息到达服务器

数据消息中的 Distance 字段可由请求消息中的 Hop 字段直接获得。此后, 在数据消息返回的过程中, Distance 字段保持不变, 因此沿途缓存节点均能从数据消息中获得。如算法 1 所示。

#### 2) 请求消息命中缓存

当请求消息命中缓存时, 数据消息中的 Benefit 字段可直接从 CS 表中 Benefit 字段读出此值。参见算法 2 中 1)~3) 行。此后, 在数据消息返回的过程中, Benefit 字段逐跳加 1, 因此沿途缓存节点均能从数据消息中获得。

#### 3) 数据消息到达 PIT 表

相应接口的数据消息中 Distance 数值等于 PIT 表中 Hop 字段值加上原数据消息的 Benefit 字段值。参见算法 3 第 7) 行。

### 5.4.3 PIT 聚合的影响

PIT 由于具有一定的聚合作用, 在数据分流的前期, 沿途缓存式 (1) 中 Distance 值只能有一个 (如图 1 中  $S-R_4-R_5$  段), 而可能对应多个用户 ( $u_2$ 、 $u_3$ 、 $u_4$ ) 和不同的 Distance (3、5、4), 无法保证每个请求在返回数据时均能严格的按照式 (1) 缓存内容。因此, 在上游公共路径阶段, 式 (1) 中 Distance 是按照首次返回的请求计算; 在分支之后阶段, 是根据不同的 Distance 值, 严格按照式 (1) 进行计算。此外, 当下游节点有多个请求消息同时到达时, 选择路径长度最长的请求向上转发, 从而保证尽量使 Distance 选取各请求消息中的最大值。

#### 算法 1

/\*服务器收到请求消息后的处理算法\*/

#### 1) 形成数据消息:

将请求消息的 Hop 赋值给数据消息中 Distance 字段;

将数据消息中 Benefit 字段初始化为 1;

#### 2) 返回数据消息。

#### 算法 2

/\*缓存节点收到请求消息后的处理算法\*/

#### 1) IF (缓存命中) /\*查 CS 表\*/

2) 形成数据消息 数据消息中的 Benefit 重置为 CS 中预先存储值 将请求消息中 Hop 与 CS

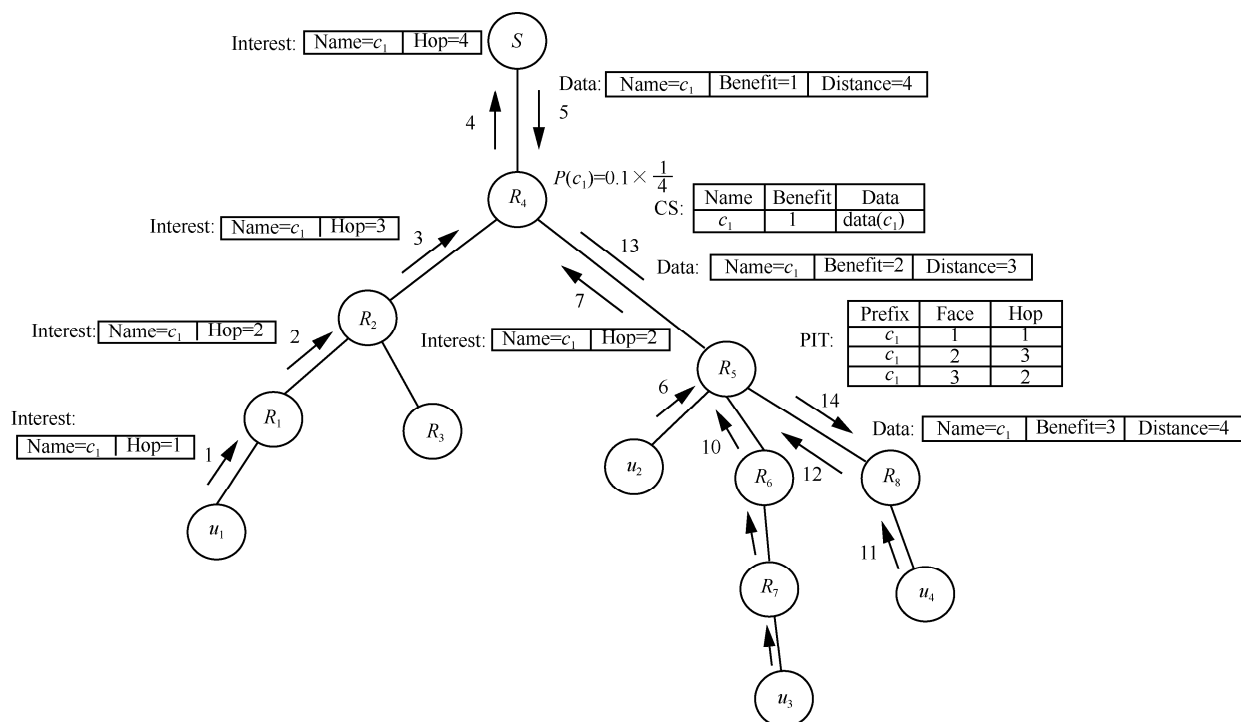


图1 案例示意

中 Benefit 之和赋值给 Distance;

- 3) 返回数据;
- 4) Else /\*缓存不命中, 查 PIT 表\*/
- 5) IF (没有匹配的 PIT 表项)
- 6) 创建一个新的 PIT 表项;
- 7) Else (有匹配的 PIT 表项)
- 8) 将请求消息到达接口加入到请求

接口列表;

- 9) 同时将请求消息中的 Hop 数值记

录在相应

接口的 hop\_temp 字段;

- 10) End IF
- 11) End IF
- 12) 查 FIB 表进行转发; /\*查 FIB 表\*/

### 算法 3

1) 结合数据消息的内容, 根据式 (1) 的概率决策是否缓存当前内容;

- 2) IF (需要缓存该内容)

3) 在 CS 表中存储内容, 同时记录数据消息中的 Benefit 值;

- 4) End IF

- 5) FOR (对于 PIT 每个表项)

6) 将返回的数据消息中 Benefit 字段加 1 后, 复制到新数据消息的 Benefit 字段;

7) 将 PIT 表中相应 Hop 值与 CS 表中 Benefit 值相加, 生成新数据消息中 Distance 字段;

- 8) 将数据从相应接口向下转发;

- 9) End FOR

同时如果此后再有更长路径的请求消息到达, 则无需再向上转发。该方法同理可适用于多重 PIT 聚合的情况。可见, 本方案需要缓存间的相互协作, 本质上属于一种缓存间的协作机制。

#### 5.4.4 具体案例

下面结合具体实例对 PCP 方案进行说明。如图 1 所示, 假设用户  $u_1 \sim u_4$  依次向网络请求内容块  $c_1$ , 其内容相对热度为 0.1。用户  $u_1$  发送请求消息, 其中携带 Hop 以记录沿途跳数。当请求消息到达服务器  $S$  后, 服务器将 Hop(=4) 值赋给数据消息的 Distance 字段。当数据消息到达  $R_4$  时, 此时借助数据消息中 Benefit 数值 1(逐跳加 1)和 Distance 数值 4, 缓存节点  $R_4$  可以计算缓存内容  $c_1$  的概率为  $0.1 \times \frac{1}{4} = 0.025$ 。假设此时  $R_4$  对内容  $c_1$  进行了缓存,

在缓存内容时节点  $R_4$  同时需要将相应的 Benefit 值 1 记录在 CS 表中。此后, 用户  $u_2$  发送请求消息到达  $R_5$  时, 将 Hop 字段数值 1 记录在 PIT 的 Hop 字段中, 由于是内容  $c_1$  的第一个请求, 请求消息继续向上转发到  $R_4$ 。在  $R_4$  缓存命中后, 需要返回数据消息。在形成数据消息时, Benefit 字段值等于 CS 表中 Benefit 数值 1 加 1, 数值为 2; Distance 字段值等于请求消息中 Hop 数值 2 加 CS 表中 Benefit 数值 1, 数值为 3。这样即使在缓存节点命中内容, 也同样可以获取 Benefit 和 Distance 值, 以帮助缓存节点进行存储决策。与此同时, 假设用户  $u_3$  和  $u_4$  相继发送针对  $c_1$  的请求消息, 并在 PIT 表中形成类似的条目。当数据消息到达节点  $R_5$  时, 针对用户  $u_4$  为例, Benefit 数值加 1 为 3, Distance 等于请求消息中 Hop 值 2 加 PIT 表中相应 Hop 值 2, 数值为 4。同理, 针对用户  $u_3$  的数据消息中, Benefit 为 3, Distance 为 5。由此可见, 在 PIT 汇聚节点的上游节点中缓存决策概率相同, 其中, Distance 值由第一个请求消息决定; 但在 PIT 汇聚节点的下游节点中缓存决策概率会有所区分, 根据各请求的 Distance 不同而有所差异。

## 6 仿真实验

为对 PCP 算法进行性能评价, 本文对主流 ndnSim<sup>[19]</sup>仿真器进行修改, 并开展仿真实验。将 PCP 与以下几种代表方案进行对比, 其中替换算法均采用 LRU 算法。

### 6.1 对比方案

现有代表性的方案有如下几种。

- 1) LCE: 传统的内容处处缓存方案<sup>[1]</sup>。

2) LCD: 内容命中后, 向请求方向的下一跳存储相应内容<sup>[6]</sup>。

3) MCD: 内容命中后, 向请求方向的下一跳存储相应内容, 并同时删除原内容<sup>[6]</sup>。

4) Prob: 每个缓存节点以固定概率缓存途经的内容<sup>[20]</sup>。

5) ProbCache: 内容放置的概率与缓存收益以及内容存储能力成正比<sup>[7]</sup>。

### 6.2 评价指标

为了便于评价 PCP 缓存方案的性能, 在此定义一些相关评价指标。

- 1) 缓存服务率

缓存服务率(CSR, cache service ratio)反映请求

由缓存而不是源服务器响应的概率。缓存命中率越高,缓存系统的效率就越高。

$$CSR = \frac{Hit\_count}{Request\_count}$$

其中,  $Hit\_count$  表示请求消息命中缓存的总次数,  $Request\_count$  表示用户发出的所有请求的数量。该指标表示缓存系统的服务效率,同时也可表示缓存缓解服务器压力的效果。

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

$$单个缓存命中率 CSR_c = \frac{Hit\_count_c}{Request\_count_c}$$

其中,  $Hit\_count_c$  表示缓存  $c$  请求命中的次数,  $Request\_count_c$  表示缓存  $c$  收到的所有请求数。

$$平均缓存命中率 ACHR = \frac{\sum_{c=1}^C CSR_c}{C}$$

其中,  $C$  表示缓存总数。

3) 平均访问延迟率 AADR(average access delay ratio)

平均访问延迟表示从用户发出请求消息到返回内容所经历的平均延迟。本指标反映缓存服务的响应速度。为方便起见,此处以跳数为单位衡量访问延迟。

$$AADR = \frac{\sum_{i=1}^N s\_c\_hop_i}{\sum_{i=1}^N s\_hop_i}$$

其中,  $\sum_{i=1}^N s\_c\_hop_i$  表示请求从缓存和服务器获取

内容的跳数总和,  $\sum_{i=1}^N s\_hop_i$  表示没有缓存时,所有请求均从服务器获取内容所需要的跳数总和。

### 6.3 实验参数设置

参考现有研究方法,实验拓扑采用  $k$ -ary 树的结构<sup>[21]</sup>,其中  $k=4$ ,每个父节点拥有 0~4 个子节点,树的层数为 2~6,默认值为 5,在 5 层的  $k$ -ary 树中共有 79 个节点,其中,51 个叶子节点为请求者,根节点为内容源,具体拓扑如图 2 所示。实验中假设内容块大小相同,用户请求模式服从 Zipf 分布,其他的主要参数如表 1 所示。实验仿真进行 60 s,取仿真开始 10 s 后的数据作为实际处理的数据。

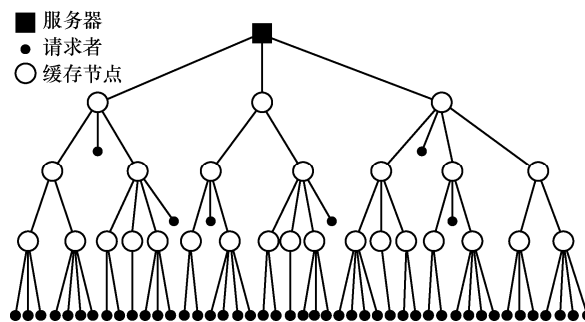


图2 实验拓扑结构

### 6.4 实验结果

本节给出实验结果,为了观察网络性能受某个参数的影响,以下各组实验中每次只让单个参数发生变化,其他参数保持不变,其默认取值如表 2 所示。

表2 实验参数

参数	默认值	变化范围
树层数	5	2~6
内容数量	10 000	4 000~40 000
缓存容量	40	25~55
请求速率	200 req/s	100~400
访问模式	Zipf: $\alpha=1.0$	0.7~1.2

#### 6.4.1 Zipf 分布参数的影响

目前,人们普遍认为用户对内容的请求模式是服从 Zipf 分布的<sup>[22]</sup>,对应不同应用 Zipf 分布的参数也不相同。本节中通过改变 Zipf 分布的参数,来研究请求模式对于缓存系统性能的影响。图 3 所示是 Zipf 参数( $\alpha$ )从 0.7 到 1.2 变化对缓存性能影响的实验结果。

图 3 可以看到随着 Zipf 参数的增大,各种缓存机制在缓存服务率、缓存命中率、平均访问延迟率方面都有一定的改善,这是符合缓存系统利用内容请求的时间局部性原理的。同时也可以看到在这 3 项指标中,PCP 策略的表现都是最优的,这也说明了 PCP 策略获得性能优势。

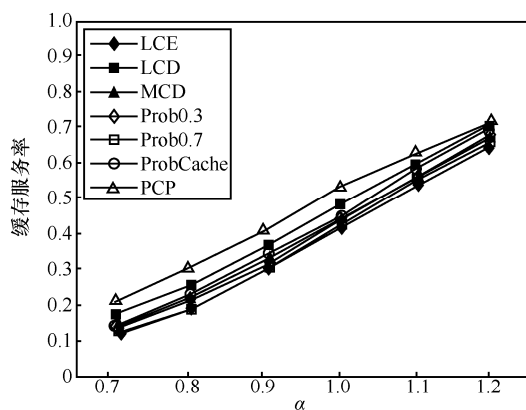
#### 6.4.2 拓扑结构的影响

本节中,改变实验拓扑结构树的层数,层数范围是 2~6,得到一组相应的实验结果。

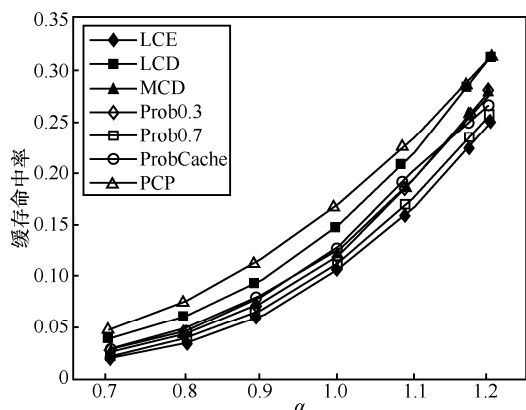
图 4 中的结果显示随着拓扑深度的增长,各项指标表现出了不同的走向。图 4(a)反映了随着深度增加,缓存的服务率逐渐增高,而在图 4(b)中看到缓存的命中率却逐步降低,这是由于随着拓扑深度的增加,路由节点数量增多,相应缓存节点的数量也增多,缓存的绝对空间增大,因此



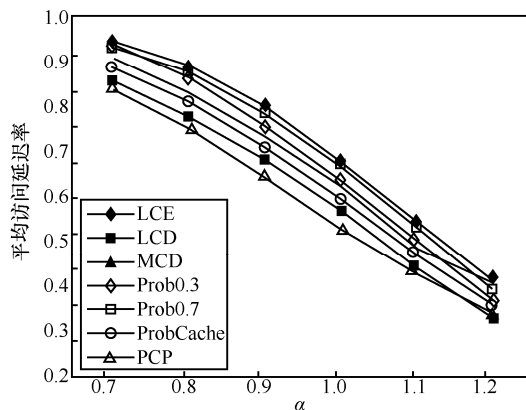
缓存服务率呈上升趋势。同时由于节点数的增多,请求也相应增加,对内容的需求量也增大,而对于单个缓存节点而言,其命中率有所下降,从而整体的缓存平均命中率也呈下降趋势。在图 4(c)中看到,用户请求的平均访问延迟率总体呈下降趋势。结合以上 3 项指标可以看出,缓存命中率与缓存服务率、平均访问延迟率等指标并无明确的相关性,因此仅考虑单一指标不足以系统地认识缓存系统。



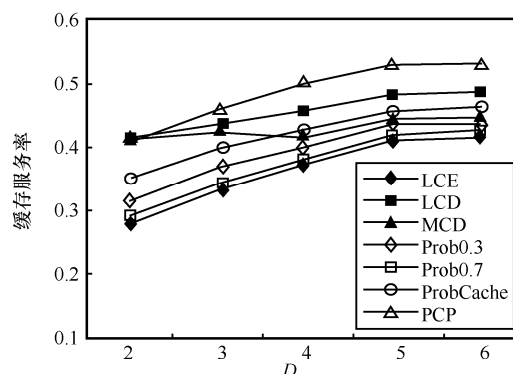
(a) 缓存服务率



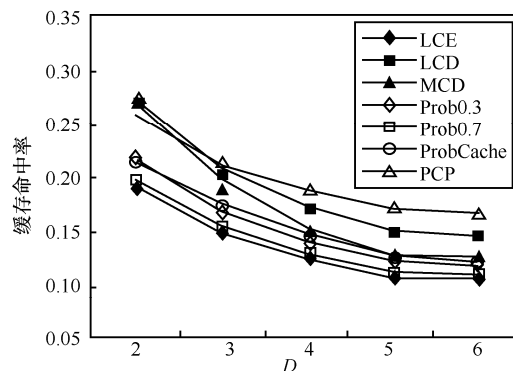
(b) 缓存命中率



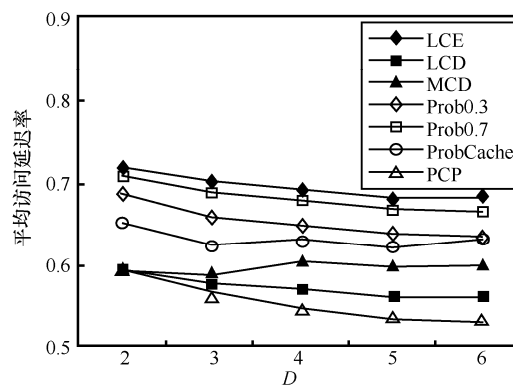
(c) 平均访问延迟率

图 3 Zipf 参数  $\alpha$  的影响

(a) 缓存服务率



(b) 缓存命中率



(c) 平均访问延迟率

图 4 拓扑深度  $D$  的影响

#### 6.4.3 请求速率的影响

本节研究在不同请求速率下各种策略的实际性能,速率的变化范围是每个请求节点每秒发送 100~400 个请求。

从图 5 中可以看出,随着请求速率的增长,除 PCP 之外的其他策略,在缓存服务率、缓存命中率、平均访问延迟率方面无明显变化,而 PCP 策略在以上 3 项指标上都随着请求速率增加有一定改善,并且优于其他各项策略,这表明在随着请求速率的增长,PCP 策略的性能仍有上升空间。

#### 6.4.4 缓存容量的影响

本节同时也在缓存容量变化的情况下,对各种

策略进行性能对比。每个缓存的容量变化范围是 25~55 个内容块。

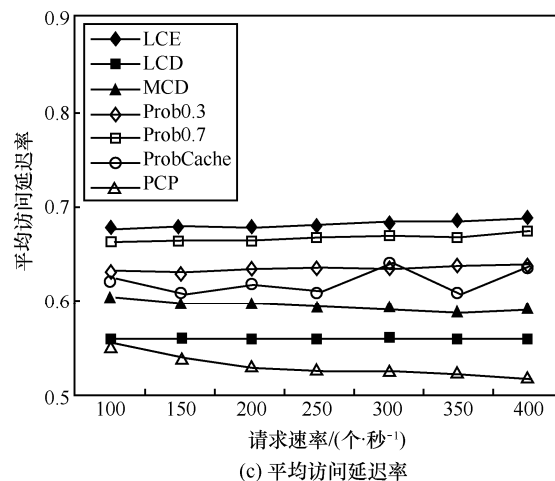
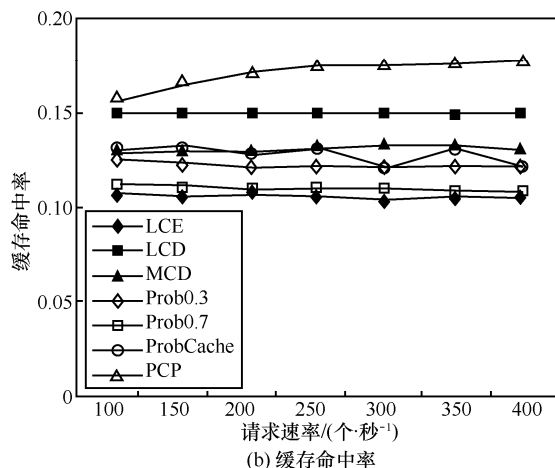
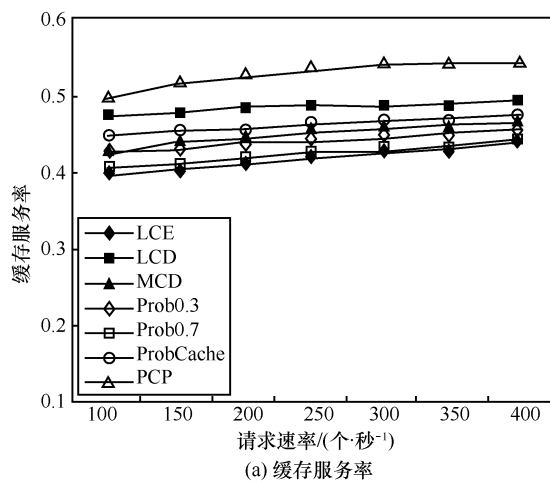


图5 请求速率的影响

从图6中可以看出,随着缓存容量的增长,所有策略,在缓存服务率、缓存命中率、平均访问延迟率方面均有所改善。这是由于缓存容量的增长可以更好地发挥缓存的作用。显而易见,PCP策略在

以上3项指标上均优于其他策略。这是由于PCP策略能够兼顾内容热度和缓存放置收益,从而能充分提高缓存空间利用率和缓存服务效率。

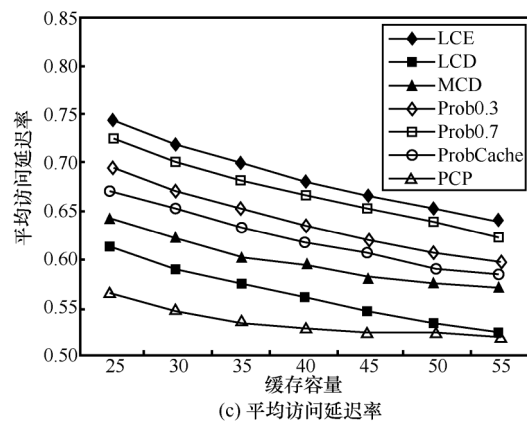
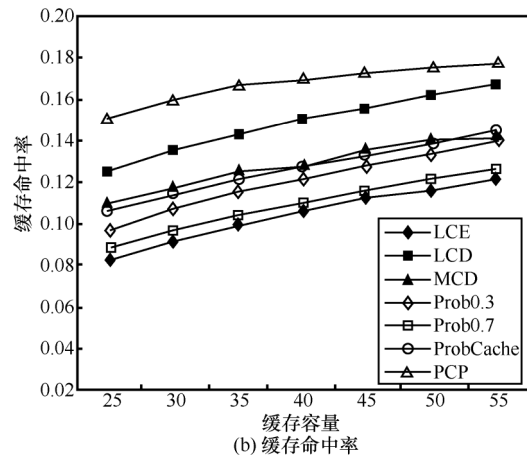
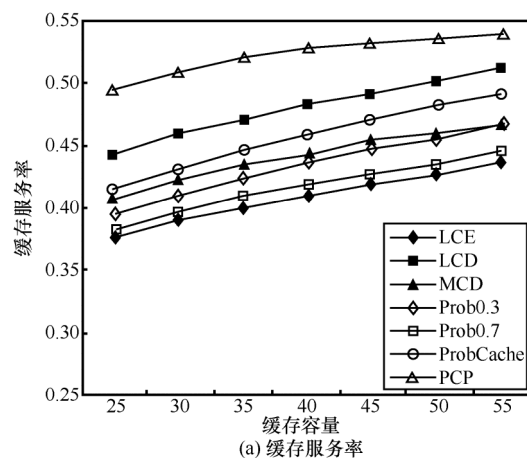


图6 缓存容量的影响

#### 6.4.5 PCP 算法开销

1) 通信开销:虽然PCP方案在请求消息中增加一个字段,在数据消息中增加2个字段,这些字段并没有增加太多额外开销,仅仅在消息中携带了一些内容,由此增加的通信开销是极为有限的,即使加入这些通信开销,PCP带来的流量节省依然明

显。另外在 CS 表和 PIT 表各增加的表项,在相应查表操作时不会增加太多开销。

2) 计算开销:概率计算是本算法中一个核心内容,下面对算法中需要的表项操作以及概率计算操作等进行了统一提取,并计算其性能,具体结果数值如表 3 所示。数据是在 2.66 GHz 的 Intel Core(TM) 2 Quad CPU Q8400 及 2 GB 内存的 PC 机上测量得到,可以看出计算开销较小,因此 PCP 完全可满足高速网络线速转发需求。

表 3 PCP 的计算开销

计算次数	总耗时/ms	平均耗时/ $\mu$ s
1 000 000	7 410	7.410
4 000 000	31 720	7.930
7 000 000	56 470	8.067
10 000 000	82 750	8.275

## 7 结束语

本文提出了一种基于概率存储内容的缓存方法 PCP,通过缓存节点合理放置缓存内容,提高和优化 ICN 缓存性能。在设计内容放置时,综合考虑内容热度和缓存放置收益,借助缓存间相互协作实现缓存内容的合理放置。实验结果表明,PCP 在缓存服务率、缓存命中率、平均访问延迟率等方面相比于现有方案均具有明显优势,同时额外通信、计算开销较小。

## 参考文献:

- [1] Cisco visual networking index: forecast and methodology, 2012~2017[EB/OL].[http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white\\_paper\\_c11-481360\\_ns827\\_Networking\\_Solutions\\_White\\_Paper.html](http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html).
- [2] JACOBSON V, SMETTERS D K, THORNTON J D, et al. Networking named content[C]//The ACM CoNEXT. ACM, c2009: 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. IEEE, c2010: 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, 2010, 66(3):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] 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. ACM, c2012: 55-60.
- [8] 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. IEEE, c2012: 316-321.
- [9] 霍如, 刘江, 黄韬, 等. 基于相关性概率的信息中心网络协作缓存策略[J]. 北京邮电大学学报, 2015, 38(1):16-20.
- [10] HUO R, LIU J, HUANG T, et al. Cooperative caching strategy based on correlation probability in information centric networking[J]. Journal of Beijing University of Posts and Telecommunications, 2015, 38(1): 16-20.
- [11] ROSENSWEIG E J, KUROSE J. Breadcrumbs: efficient, best-effort content location in cache networks[C]//The IEEE INFOCOM. IEEE, c2009: 2631-2635.
- [12] ZHU Y, CHEN M, NAKAO A. Conic: content-oriented network with indexed caching[C]//The IEEE INFOCOM 2010 Workshops. IEEE, c2010: 1-6.
- [13] LI Z, SIMON G. Time-shifted TV in content centric networks: the case for cooperative in-network caching[J]//The 2011 IEEE ICC. IEEE, c2011: 1-6.
- [14] MING Z, XU M, WANG D. Age-based cooperative caching in information-centric networks[C]//IEEE INFOCOM 2012 on Computer Communications Workshops. IEEE, c2012: 268-273.
- [15] WANG S, BI J, WU J, et al. Could in-network caching benefit information-centric networking[C]//AINTEC. c2011: 112-115.
- [16] LIU W X, YU S Z, CAI J, et al. Scheme for cooperative caching in ICN[J]. Journal of Software, 2013, 24(8):1947-1962.
- [17] WANG L, HOQUE A, YI C, et al. OSPFN: an OSPF based routing protocol for named data networking[R]. University of Memphis and University of Arizona, Tech Rep, 2012.
- [18] MAHMUDUL A K M, SYED O A, ADAM A, et al. NLSR: named-data link state routing protocol[C]//The ACM SIGCOMM Workshop. c2013: 15-20.
- [19] HU X Y, GONG J, CHENG G, et al. Enhancing in-network caching by coupling cache placement, replacement and location[C]//ICC. c2015: 7301-7307.
- [20] AFANASYEV A, MOISEENKO I, ZHANG L. ndnSIM: NDN simulator for NS-3, NDN[R]. Technical Report NDN-0005, 2012.
- [21] LAOUTARIS N, SYNTILA S, STAVRAKAKIS I. Meta algorithms for hierarchical Web caches[C]//IPCCC. c2004: 445-452.
- [22] CHAI W K, HE D, PSARAS I, et al. Cache Less for more in information-centric networks[C]//The NETWORKING 2012. Springer Berlin Heidelberg, c2012: 27-40.
- [23] LEE B, PEI C, LI F, et al. Web caching and Zipf-like distributions: evidence and implications[C]//The IEEE INFOCOM. c1999: 126-134.

## 作者简介:



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。

李俊 (1968-), 男, 安徽桐城人, 中国科学院计算机网络信息中心研究员、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。

智江 (1983-), 男, 山西太原人, 中国科学院研究生院博士生, 主要研究方向为下一代互联网关键技术。