

## 基于收益感知的信息中心网络缓存机制

陈龙, 汤红波, 罗兴国, 柏溢, 张震

(国家数字交换系统工程技术研究中心, 河南 郑州 450002)

**摘要:** 针对信息中心网络 (ICN) 内置缓存系统中的海量内容块流行度获取和存储资源高效利用问题, 以最大化节省内容访问总代价为目标, 建立针对内容块流行度的缓存收益优化模型, 提出了一种基于收益感知的缓存机制。该机制利用缓存对请求流的过滤效应, 在最大化单点缓存收益的同时潜在地实现节点间协作和多样化缓存; 使用基于布隆过滤器的滑动窗口策略, 在检测请求到达间隔时间的同时兼顾从源服务器获取内容的代价, 捕获缓存收益高的内容块。分析表明, 该方法能够大幅压缩获取内容流行度的存储空间开销; 仿真结果表明, 该方法能够较为准确地实现基于流行度的缓存收益感知, 且在内容流行度动态变化的情况下, 在带宽节省和缓存命中率方面更具优势。

**关键词:** 信息中心网络; 内容路由; 缓存机制; 内容流行度

**中图分类号:** TP393

**文献标识码:** A

## Gain-aware caching scheme for information-centric networking

CHEN Long, TANG Hong-bo, LUO Xing-guo, BAI Yi, ZHANG Zhen

(National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China)

**Abstract:** The in-network caching system of information-centric networking had to deal with the popularity of huge number of content chunks and make efficient usage of storage resources. A content popularity based caching gain optimization model aimed to get maximum reduction of content retrieve cost was established, and a gain-aware caching scheme was proposed. By utilizing filtering effect of cache to request flow, this scheme achieves caching cooperation and diversity potentially while maximizing caching gain of every single node. Bloom filter based sliding window strategy captures the content chunks with high caching gain according to request arrival interval and retrieval cost from the source. Analysis shows that the method can drastically reduce memory consumption caused by popularity monitoring. The simulation results illuminate that this scheme is well aware of content popularity based caching gain, and gets better bandwidth saving and cache hit ratio when content popularity is changing dynamically.

**Key words:** information-centric networking, content-based routing, caching scheme, content popularity

### 1 引言

随着互联网应用的飞速发展, 网络流量高速增长, 信息获取逐渐成为用户的核心需求。为应对这一变化, 并克服 TCP/IP 协议架构存在的移动性、扩展性、安全性等问题, 研究人员近年提出了信息

中心网络 (ICN, information-centric networking) 这一类新型网络体系架构<sup>[1]</sup>。ICN 将网络通信模式从以端为中心转变为以信息为中心, 直接对内容进行统一标识并基于内容标识进行定位、路由和传输, 使用网内缓存 (in-network caching) 系统加速内容分发, 将网络打造成为信息传输、存储和服务的一

收稿日期: 2015-10-26; 修回日期: 2016-04-08

通信作者: 汤红波, 1480236389@qq.com

基金项目: 国家自然科学基金资助项目 (No.61372121); 国家高技术研究发展计划 (“863”计划) 基金资助项目 (No.2014AA01A701); 国家科技重大专项基金资助项目 (No.2013ZX03006003-009)

**Foundation Items:** The National Natural Science Foundation of China (No. 61372121); The National High-Tech Research and Development Program (863 Program) of China (No.2014AA01A701); The National Science and Technology Major Project of China (No.2013ZX03006003-009)

体化平台,被认为是最有潜力的下一代互联网体系结构之一<sup>[1]</sup>。其中,网内缓存作为 ICN 的关键技术之一,利用网络节点中的存储空间缓存内容副本,就近响应用户请求,可以有效减少冗余流量,缓解网络带宽压力,降低用户内容获取时间。虽然缓存相关优化技术已经在 Web 缓存、内容分发网络(CDN, content delivery network)和对等网络(P2P, peer-to-peer)中得到了较为广泛的研究,但 ICN 网内缓存所呈现出的透明化、泛在化和细粒度化等新特征致使传统的缓存方法无法适用或需要重新考量<sup>[2]</sup>。

NDN (named data networking)<sup>[3]</sup>等 ICN 架构实例默认采用沿途全部缓存(CE<sup>2</sup>, cache everything everywhere)方式,将所请求的内容对象在返回路径上所有节点都进行存储,但这种策略致使节点缓存内容趋于同质化,内容频繁替换更新,缓存缺失概率较大。为提升缓存空间利用效率,研究者们提出了若干缓存方案,从加强节点间协作出发提高缓存内容的差异性,基于用户访问特征提高缓存内容的针对性,在一定程度上提高了缓存系统的性能,但仍然面临以下两方面的挑战。

1) 内容流行度获取问题。许多缓存方案依据内容的流行程度进行缓存放置,更多地缓存用户访问频率较高的内容,但是这些方案并未给出获取内容流行度的具体方法,而 ICN 缓存单元为细粒度的内容块(chunk),由于内容块数量庞大,且在不同的缓存节点呈现不同的流行度特征,内容块访问频率统计在存储和查找开销方面存在较大挑战。

2) 协作缓存的可扩展性问题。为实现差异化缓存,邻域协作缓存方案或需要节点间通告状态信息,或需要预先建立较为固定的协作域,存在状态信息交互量大、协作域维护复杂等问题;路径协作缓存方案通过用户消息捎带实现传输路径中节点间的缓存信息交互,但扩展用户消息不但增加了实现和部署的复杂度,还引入了节点间信任问题。

针对上述问题,本文建立以最大化降低内容访问总代价为目标的缓存收益优化模型,提出一种基于收益感知的缓存机制(GACS, gain-aware caching scheme)。在协作缓存方面,该机制利用缓存对请求流的过滤效应,以最大化单点缓存收益为目标,在单点缓存决策的同时潜在地实现节点间协作,具有良好的可扩展性;统筹考虑内容

流行度获取和缓存决策过程,使用基于布隆过滤器(Bloom filter)的滑动窗口策略,通过比较请求到达间隔感知内容块的流行度高低,并依据从源服务器获取内容的代价设定窗口长度,实现高缓存收益率内容块的捕获。最后,通过复杂度分析和仿真实验,说明该机制具有额外存储空间消耗低、访问频率特征变化跟踪能力强、缓存空间资源利用效率高的特点。

## 2 相关工作

ICN 中,由于内容流量的多样性和缓存执行的线速要求,无法通过先验的拓扑和流量信息以及线下的计算进行内容的预先放置。ICN 采用“请求—响应”的内容获取模式,缓存发生在内容返回时,从而缓存机制可以分为缓存决策和缓存替换两部分。缓存决策机制决定是否存储当前通过的内容对象,当缓存空间已满且又有新的内容对象进入缓存时,缓存替换算法需要选择一个缓存内容被替换掉。由于在线缓存决策是 ICN 给缓存系统提出的新要求,所以 ICN 缓存优化技术给予了缓存决策更多的关注。

依据节点间协作是否需要建立协作域和交互状态信息,ICN 缓存策略可以大致分为显式协作和隐式协作两类<sup>[2]</sup>。在显式协作缓存机制中,较为直接的一类方案<sup>[4]</sup>是通过在协作域内节点之间交互访问模式、缓存状态等信息,提高协作域内的缓存多样性,但是交互开销是具有较强动态性的 ICN 缓存系统难以承受的。基于散列的协作缓存机制<sup>[5,6]</sup>通过散列函数确定哪一个节点缓存该内容对象,避免了复杂的协作算法,但是需要预先划分协作域并分配一致的散列函数,节点加入退出较为复杂。

在隐式协作缓存机制中,节点间不需要或较少地交互信息即可完成缓存决策,在执行速度和复杂度方面更具优势。按照是否需要事先获知准确的内容流行度信息,可以将隐式协作缓存机制分为以下两类。

### 1) 潜在考虑内容流行度的缓存机制

LCD (leave copy down)<sup>[7]</sup>缓存机制只在缓存命中节点的下一跳节点存储应答内容;MCD(move copy down)<sup>[7]</sup>机制将缓存内容对象从命中节点移动到下一跳节点存储,进一步降低了缓存冗余。文献[8]提出了一种随机缓存机制(RCOne),在沿途传输路径上随机选择一个节点存储内容对象。基于加权概率的缓存

机制 (ProbCache)<sup>[9,10]</sup>依据节点在数据传输路径中的位置和路径节点缓存容量进行概率缓存决策,使内容以更大的概率缓存在距离请求者更近的节点上。文献[11]提出了一种关联缓存决策机制 (WAVE),节点通过设置标记为后续节点提供缓存建议,当以文件为单位的请求次数增加时,以指数增长方式增加传输路径上缓存该文件内容块的个数。文献[12,13]分别根据缓存节点的介数和在社团中的重要度进行缓存决策。该类方案决策过程较为简单,但是缺乏准确的内容访问频率信息,存在内容复制到边缘的速度较慢、内容在边缘节点过分竞争、不同传输路径的内容无序争用缓存空间等问题。

2) 基于内容流行度信息的缓存机制

文献[14]提出了基于缓存寿命 (age) 的机制,依据内容流行等级和存储节点的位置来计算内容的缓存寿命,内容流行度越高,距离源服务器越远,缓存驻留时间越长。文献[15]提出了基于流行度的概率缓存机制 (prob-PD),根据内容流行度和距离内容源的距离进行概率缓存决策。刘外喜等<sup>[16]</sup>提出了一种选择性缓存机制,根据用户的潜在需求和内容的流行规律降低缓存冗余。文献[17]提出了一种基于收益最大化的分布式缓存机制 (MAGIC),通过综合考虑内容流行度和缓存内容所节省的传输跳数,计算缓存收益和替换损失,在请求消息传输过程中收集所经过节点中该内容块对应的最大缓存收益增加值,内容块回传时,找到缓存收益增幅最大的节点存储该内容块,但是对于高收益内容块,可能需要多次请求后才会被复制到网络边缘。Badov 等<sup>[18]</sup>提出了一种拥塞感知的缓存和查找机

制,该机制在感知路径拥塞的基础上,有针对性地缓存对缓解拥塞更有帮助的内容,并在请求消息转发时,优先考虑带宽负载较低的链路。表 1 给出了相关缓存机制的特点对比,该类方案在缓存命中率 and 带宽节省方面都有较好的性能表现,但是大都假设内容流行度是已知的参量,并没有明确描述内容流行度信息是如何由路由器获得和维护的。

ICN 缓存系统中,除了用户内容访问特征固有的时变性,由于缓存的网络结构,请求到达过程还与上游节点的缓存状态有关<sup>[19]</sup>,从而内容请求到达率具有较强的动态性,静态线下的流行度统计显然不能很好地满足缓存决策的需求。对于在线流行度感知,一种直接的方法是使用散列表统计内容的访问次数,但在需要处理高速请求流和海量内容请求的 ICN 环境中,这种方法存在的存储开销是难以承受的<sup>[20]</sup>。一些方法通过粗粒度的内容访问频率统计降低复杂度,具体方法包括按照流行度将内容划分为若干等级<sup>[21,22]</sup>、使用内容文件的流行度代替内容块的流行度<sup>[23]</sup>等。但是,一方面,由于用户本身可能对同一内容文件的内容块表现出不同程度的兴趣,另一方面,同一内容文件的内容块可能缓存在不同节点上,由于上游的缓存命中,使这些内容块在不同节点上表现出不同的流行度,这些方法忽视了这些差异性,影响了细粒度缓存管理的性能。文献[20]提出了基于布隆过滤器的在线内容流行度捕获算法,统计固定周期内的内容请求到达次数,但对于不太流行的内容,存在较大误差,另外,该方法没有给出统计周期的设置方法,造成流行度统计缺少针对性,在存储和计算开销方面仍然存在优化

表 1 缓存机制特点比较

缓存机制	缓存决策依据	是否需要信息交互	是否依赖流行度信息	缓存命中率
CE <sup>[23]</sup>	全部缓存	否	否	低
LCD、MCD <sup>[7]</sup>	缓存命中事件	是	否	中
RCOne <sup>[8]</sup>	缓存节点随机选择	是	否	中
ProbCache <sup>[9,10]</sup>	缓存容量、节点位置	否	否	中
WAVE <sup>[11]</sup>	文件访问次数	是	否	中
Betw <sup>[12]</sup>	节点介数	否	否	中
Age-based <sup>[14]</sup>	内容流行度、节点位置	是	是	高
Prob-PD <sup>[15]</sup>	内容流行度、节点位置	否	是	高
Liu's <sup>[16]</sup>	内容的流行规律	是	是	高
MAGIC <sup>[17]</sup>	内容流行度、节点位置	是	是	高
Badov's <sup>[18]</sup>	内容流行度、链路状态	是	是	高

空间。

为降低流行度获取给缓存系统带来的负担，本文重新考虑缓存机制对内容流行度的感知需求，将缓存放置建模为缓存收益优化问题，使用带权的内容流行度比较，避免对内容访问频率的精确统计；使用存储高效的布隆过滤器筛选访问间隔较短的内容块，在缓存收益感知的同时实现内容块在缓存网络中的合理放置。

### 3 缓存收益优化模型

将 ICN 缓存系统建模为无向图  $G(V, L)$ ，其中， $V$  为所有节点的集合，由内容源服务器集合  $S$  和缓存节点集合  $N$  构成，即  $V = S \cup N$ ， $L$  表示所有链路的集合。记  $K$  为内容集合，为实现细粒度缓存管理，这里  $K$  中的元素即为内容块（chunk），假定内容块大小相同，任意内容块  $k$  ( $k \in K$ ) 拥有唯一且固定的内容源服务器  $s(k) \in S$ ，内容源服务器直接与一个缓存节点连接。

表 2 符号说明

符号	含义
$N$	缓存节点集合
$S$	内容源服务器集合
$K$	网络中的内容块集合
$\gamma_n^k$	节点 $n$ 上内容块 $k$ 的外生请求到达率
$\lambda_n^k$	节点 $n$ 上内容块 $k$ 的请求到达率
$\lambda_n$	节点 $n$ 上内容请求总到达率
$h_s^k$	时隙 $s_t$ 中内容块 $k$ 的命中次数
$\delta_n^k$	缓存节点 $n$ 上是否存在有内容块 $k$
$s(k)$	内容块 $k$ 的源服务器
$c_n^k$	节点 $n$ 从 $s(k)$ 获取内容块 $k$ 所产生的代价
$v_n$	节点 $n$ 能够缓存内容块的数量
$K_n$	节点 $n$ 缓存内容块的集合

内容请求消息可以从网内任何缓存节点进入网络，记  $\gamma_n^k$  为由节点  $n$  进入缓存网络的对内容块  $k$  的请求到达率，即外生请求到达率， $\lambda_n^k$  为节点  $n$  收到的内容块  $k$  的请求到达率。请求消息按照路由机制转发到内容源节点或缓存有该内容的节点，相匹配的数据应答消息沿请求路径返回。记内容块  $k$  的访问代价  $c_n^k$  为请求消息从节点  $n$  出发，到达内容源服务器  $s(k)$ ，并从  $s(k)$  获取内容块  $k$  所产生的代价，这里取  $c_n^k$  为节点  $n$  与内容块  $k$  源服务器  $s(k)$  之间的跳数， $c_n^k$  可以通过路由协议获取，典型的有

NDN 中的 OSPFN 协议<sup>[3]</sup>。路由协议在计算转发路径、生成路由转发表的同时可以得到节点与源服务器之间的跳数。

记  $v_n$  为节点  $n$  ( $n \in N$ ) 的缓存容量，即节点能够存储内容块的数量。定义缓存决策变量集合  $\Delta$ ， $\delta_n^k \in \Delta$  表示内容块  $k$  是否存储在缓存节点  $n$  上，若存储，则  $\delta_n^k = 1$ ，否则  $\delta_n^k = 0$ 。

通过合理的缓存内容放置最小化内容的访问代价可以等效为最大化代价节省问题，即相比所有请求都从源服务器获取内容的情况，缓存命中带来的收益最大化。定义缓存收益率为单位时间内的内容访问代价节省量，则内容块  $k$  存储在节点  $n$  带来的缓存收益率  $g_n(k) = \lambda_n^k c_n^k$ ，并得到如下缓存收益率优化模型

$$\max g(\Delta) = \sum_{n \in N} \sum_{k \in K} g_n(k) \delta_n^k \quad (1)$$

$$\text{s.t.} \quad \sum_{k \in K} \delta_n^k \leq v_n, n \in N \quad (2)$$

$$\lambda_n^k = \sum_{m \in P(n,k)} \lambda_m^k (1 - \delta_m^k) + \gamma_n^k \quad (3)$$

$$\sum_{m \in N} \gamma_m^k = \sum_{n \in N} \lambda_n^k \delta_n^k + \lambda_{s(k)}^k \quad (4)$$

约束式(2)表示节点中存储的内容块数量不能超出节点容量。式(3)为路由约束，其中， $P(n, k)$  为针对内容块  $k$ ，以节点  $n$  为直接路由后继的节点集合，表示到达节点的内容请求包括外生的请求和邻居节点由于缓存未命中而向其转发的请求。约束式(4)表示内容请求只能同时由一个内容源服务器或缓存节点提供服务。

在请求消息路由转发策略给定的情况下，上述缓存放置问题为背包问题，可以通过贪婪算法求解，但是 ICN 缓存系统并不存在集中式的管理服务器，缓存算法需要分布式地部署在每一个节点上，各节点需要收集其他节点的内容请求到达信息才能计算出最优的内容缓存位置不但会引入额外的信息交互开销，增加实现的复杂度，而且所有的节点必须支持相同的缓存管理协议，引入可扩展性问题，不利于缓存机制的部署应用。

为此，令节点独立地进行缓存决策，在其他节点缓存状态未知的情况下，节点  $n$  选择存储缓存收益率最高的  $v_n$  个内容块，即对每一个节点  $n$ ，使其达到

$$\max g_n(\Delta) = \sum_{k \in K} g_n(k) \delta_n^k \quad (5)$$

虽然上述单点缓存收益优化目标并没有实现严格的全局最优，但也并不是完全无序的独立缓存决策，而是存在一定的协同效应。内容请求路径中靠近用户的节点识别并存储缓存收益率最高的内容，按照约束式(3)和式(4)，缓存对内容请求流存在过滤效果，对于已经在上游缓存的内容，后续缓存节点收到的对于这些内容的请求到达率将会降低，从而后续节点不再缓存这些内容，潜在地实现了一定程度的协同。以图 1 所示的场景为例说明这种协同效应，缓存节点  $n_1$ 、 $n_2$ 、 $n_3$  各能存储 1 个内容块，请求者通过节点  $n_1$  接入网络，对内容块的访问频率满足  $\lambda_{n_1}^{k_1} > \lambda_{n_1}^{k_2} > \lambda_{n_1}^{k_3} > \dots$ 。按照式(5)各个节点存储缓存收益率最高的内容块，节点  $n_1$  存储内容块  $k_1$ ，能够响应对内容块  $k_1$  的请求，则节点  $n_2$  上内容块  $k_1$  的请求到达率  $\lambda_{n_2}^{k_1} = 0$ ， $k_2$  成为节点  $n_2$  处缓存收益率最高的内容块，同理节点  $n_3$  缓存  $k_3$ ，从而节点间形成协同效应，缓存总收益得到优化。

为更好地说明单点缓存收益优化的有效性，对于更一般的场景，分析独立缓存决策与全局最优缓存总收益比值的下限。考虑一种最坏的情况，即节点  $n$  中存储所有内容块同时被后续邻居节点缓存，则节点  $n$  中缓存的任意一个内容块  $k_1$ ，如果被替换为内容块  $k_2$ ， $k_1$  替换后总缓存收益减少  $\lambda_n^{k_1}$ ，则系统缓存总收益的变化量为  $c_n^{k_2} \lambda_n^{k_2} - \lambda_n^{k_1}$ 。如果对节点  $n$  中的缓存内容块进行调整，系统缓存总收益的增量最大为

$$g'_n = \max_{K_1, K_2} \left\{ \sum_{k_2 \in K_2} c_n^{k_2} \lambda_n^{k_2} - \sum_{k_1 \in K_1} \lambda_n^{k_1} \right\} \quad (6)$$

其中， $K_1$  为被替换出缓存的内容块集合， $K_2$  为进入缓存的内容块集合。记节点  $n$  按照式(5)得到的单点缓存收益最大值为  $g_n^{\max}$ ，缓存系统总收益的全局最优解为  $g^{\max}$ ，则  $g^{\max} \leq \sum_{n \in N} (g_n^{\max} + g'_n)$ ，从而

$$\frac{\sum_{n \in N} g_n^{\max}}{g^{\max}} \geq \frac{\sum_{n \in N} g_n^{\max}}{\sum_{n \in N} (g_n^{\max} + g'_n)} \geq \min_{n \in N} \left\{ \frac{g_n^{\max}}{g_n^{\max} + g'_n} \right\} \quad (7)$$

记  $b_i = \frac{g_n^{\max}}{g_n^{\max} + g'_n}$ ，下面对  $b_i$  进行数值分析。为简化分析并不失一般性，假设内容块拥有同一个内容服务器，请求概率服从参数为  $\alpha$  的 Zipf 分布<sup>[24]</sup>，第  $i$  流行的内容块请求概率为  $p(i) = \frac{X}{i^\alpha}$ ， $X = (\sum_i \frac{1}{i^\alpha})^{-1}$ ，内容块总数量设置为 1 000，节点容量为 100， $\alpha$  分别取 0.7 和 1 时， $b_i$  随  $c_n^k$  的变化情况如图 2 所示， $b_i$  最低为 0.755 4，可以看出独立决策与全局最优的差距是可控的，这说明即便后续节点上存储有相同内容块，独立缓存决策带来的收益损失仍然不会太大，因为根据缓存过滤效应，独立缓存决策是一种边缘节点优先的策略，而边缘节点往往与内容服务器距离较远，对系统总收益的贡献更大。

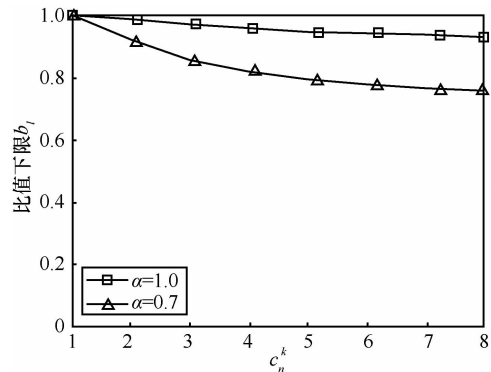


图 2 独立缓存决策与全局最优缓存总收益比值下限

### 4 基于收益感知的缓存机制

按照上节给出的优化目标进行缓存决策和替换。首先根据访问间隔时间判断内容块访问到达率的大小，得到缓存决策规则，进而基于布隆过滤器检测访问间隔，将缓存收益率更高的内容放置在节

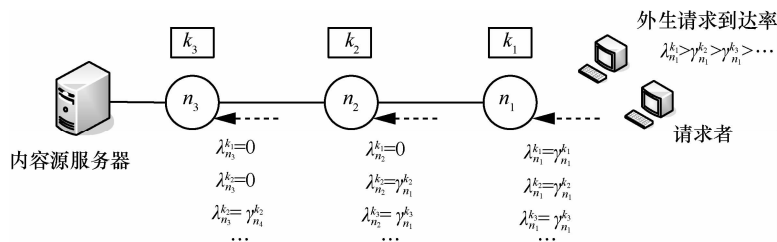


图 1 基于过滤效应的缓存协同实例

点缓存中。

### 4.1 决策规则

按照上述优化模型，缓存节点应当存储当前缓存收益率最高的  $v_n$  个内容块。注意到缓存机制并不直接需要缓存收益率的精确值，而是需要比较当前传输内容块与已缓存内容块缓存收益率大小关系。根据式(5)，记节点  $n$  缓存内容集合为  $K_n$ ，当前缓存内容中最小缓存收益率为  $g_{\min} = \min\{g_n^k, k \in K_n\}$ ，对于当前请求的内容块  $k \notin K_n$ ，有  $g_n(k) > g_{\min}$ ，即  $\lambda_n^k > \frac{g_{\min}}{c_n^k}$ ，则应当缓存内容块  $k$ 。

内容块  $k$  在节点  $n$  处的缓存收益率  $g_n(k)$  为从源服务器获取内容的代价  $c_n^k$  和请求到达率  $\lambda_n^k$  的乘积，并假定  $c_n^k$  已由路由生成机制获取，并记录在路由转发表项中，在内容请求到达时可以通过查询得到。从而，缓存收益率感知的重点落在内容请求到达率的检测上。

这里通过检测内容访问到达间隔来感知内容的请求到达率。定义节点  $n$  对内容块  $k$  的访问时间间隔序列为

$$IRG_n(k) = \{x_1^k, x_2^k, x_3^k, \dots, x_i^k, \dots\}, x_i^k = t_i - t_{i-1}, t_0 = 0 \quad (8)$$

其中， $t_i$  为内容块  $k$  第  $i$  次访问到达时刻。

由于 ICN 中内容请求到达率表现出较强的动态性，从而缓存决策算法需要对流行度的变化具有较强的敏感性，兼顾当前时刻内容块请求特征的新颖度和缓存状态的稳定性，分别考虑最近的 1 次访问间隔和最近的  $m$  ( $m \geq 2$ ) 次访问间隔，设计以下 2 个缓存决策条件

$$x_i^k < p \frac{c_n^k}{g_{\min}} \quad (9)$$

$$x_i^k + x_{i-1}^k + \dots + x_{i+m-1}^k < p \frac{mc_n^k}{g_{\min}} \quad (10)$$

其中， $p$  ( $0 < p \leq 1$ ) 为门限调节因子，下面讨论  $p$  和  $m$  的取值对检测效果的影响。

定义检测率为在内容块  $k$  平均缓存收益率大于  $g_{\min}$  的条件下，式(9)和式(10)能够同时得到满足的概率；误判率为在式(9)和式(10)得到满足的条件下，内容块  $k$  平均缓存收益率小于  $g_{\min}$  的条件下概率。按照请求概率服从参数为  $\alpha$  的 Zipf 分布生成内容访问序列，内容块数量设置为 1000，取第 100 流行内容的缓存收益为  $g_{\min}$ ， $c_n^k = 1$ ，当  $p=0.8$  时统计得到

检测率和误判率随  $m$  值的变化情况，如图 3 (a) 所示， $m=1$  表示决策条件只有式(9)，可以看出随着  $m$  值的增加误判率不断降低而检测率变化不大。图 3

(b) 给出了  $m=2, \alpha=1$  时检测率和误判率随  $p$  值的变化情况，可以看出检测率和误判率都随  $p$  值的减小而降低。统筹考虑  $p$  和  $m$  的取值，考虑到对流行度变化的敏感性以及可能带来的存储开销，取  $m=2$ ，同时通过减小  $p$  降低误判率，提高系统稳定性。为方便算法实现，避免获取  $g_{\min}$  时比较和查找等操作带来的复杂度增加，取  $p = \frac{g_{\min}}{g_n}$ ，其中，

$\tilde{g}_n = \frac{\sum_{k \in K_n} g_n^k}{v_n}$ ，为缓存内容的平均缓存收益率，则式(9)和式(10)可以转换为

$$x_i^k < \frac{c_n^k}{g_n} \quad (11)$$

$$x_i^k + x_{i-1}^k < \frac{2c_n^k}{g_n} \quad (12)$$

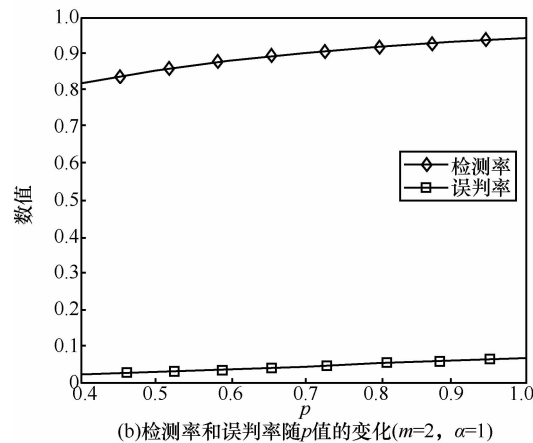
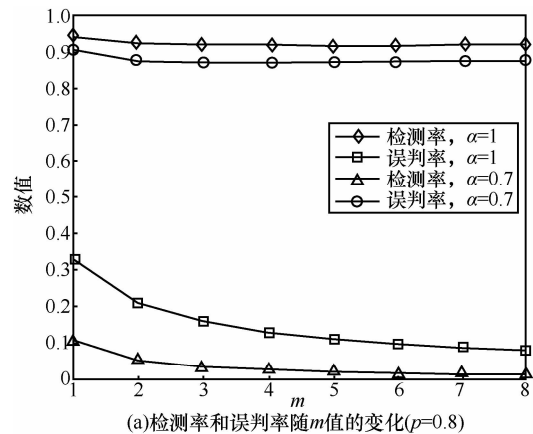


图 3 检测率和误判率统计分析

下面按照上述规则设计缓存决策算法。

### 4.2 算法设计

ICN 中由于内容块的数量巨大，存储所有内容块的访问间隔序列会带来较大的存储和查询开销。为此，引入布隆过滤器，在访问间隔检测的基础上实现缓存收益感知。布隆过滤器是一种用来检测一个元素是不是集合中成员的数据结构，在空间和时间方面优势明显，存储空间和插入/查询时间都是常数<sup>[24]</sup>。

如图 4 所示，将时间划分为连续的时隙，任意一个时隙  $s_i$  内有请求到达的内容块集合由一个布隆过滤器  $BF_i$  记录。通过查询布隆过滤器  $BF_i$ ，可以得到当前请求的内容块是否在时隙  $s_i$  内被请求过。

取  $s_i$  为  $\frac{1}{g_n}$ ，则缓存决策条件式(11)可以描述为在  $c_n^k$  个时隙内有 2 次请求到达，式(12)可以描述为  $2c_n^k$  个时隙内有 3 次请求到达。

当内容请求到达时，如果本地缓存没有命中请求，则执行缓存决策算法。假定当前时刻处于在时隙  $s_i$  内，首先根据内容块  $k$  的标识查询访问代价  $c_n^k = j$ ，然后依次查询布隆过滤器  $BF_i, BF_{i-1}, \dots, BF_{i-j}$ 。如果全部匹配失败，则决策不存储内容块  $k$ ；如果内容块  $k$  的标识匹配其中的  $BF_a (i \leq a \leq i-j)$ ，则依次查询  $BF_{a-1}, BF_{a-2}, \dots, BF_{i-2j}$ ，如果匹配任意一个布隆过滤器，则决策缓存该内容块。另外，如果内容块  $k$  不在  $BF_i$  中，则将内容块  $k$  插入到  $BF_i$ 。下面给出算法的伪代码。

BF\_Cache\_Decision(ContentName name)

返回：是否决策缓存 True/False

- 1) 通过查询 FIB 表项获取  $c_n^k$
- 2) IF Query( $BF_i, name$ )=False Then

- 3) Insert( $BF_i, name$ );
- 4) For  $j:1 \rightarrow c_n^k$  do
- 5) IF Query( $BF_{i-j}, name$ )=True Then
- 6) For  $r:j \rightarrow 2c_n^k$  do
- 7) IF Query( $BF_{i-r}, name$ ) = True Then
- 8) return True
- 9) End IF
- 10) End IF
- 11) Else
- 12) For  $j:1 \rightarrow 2c_n^k$  do
- 13) IF Query( $BF_{i-j}, name$ )=True Then
- 14) return True
- 15) End IF
- 16) return False

如果当前时隙结束，根据  $\tilde{g}_n$  设定下一个时隙的长度，动态适配消息请求速率的变化，使每个时隙请求到达总数大致相同，则内容块在每个时隙被请求的概率可以更好地匹配流行度的大小，不会被请求总速率的变化所影响，从而降低流行度比较的误差。下面说明平均缓存收益率  $\tilde{g}_n$  的获取方法。通过对时隙内每次缓存命中时获取的收益进行统计平均，可以得到每个时隙的平均缓存收益率  $\tilde{g}_n(s_i)$

$$\tilde{g}_n(s_i) = \frac{\sum_{k \in K_n} c_n^k h_{s_i}^k}{v_n s_i} \quad (13)$$

其中， $h_{s_i}^k$  为在时隙  $s_i$  中内容块  $k$  的命中次数。平均缓存收益率  $\tilde{g}_n$  由迭代产生

$$\tilde{g}_n = j\tilde{g}_n(s_i) + (1-j)\tilde{g}_n \quad (14)$$

为了兼顾平均值的新颖度和稳定性， $j$  取 0.5。

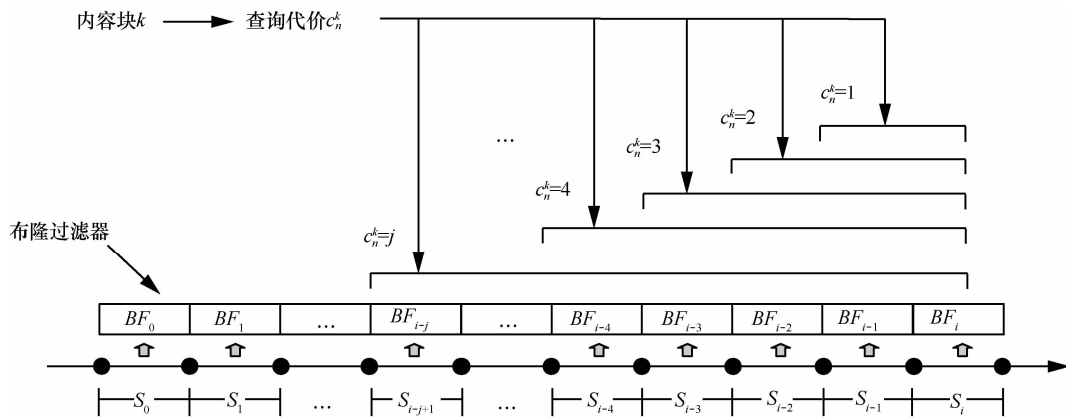


图 4 基于布隆过滤器的缓存收益检测

由于缓存空间有限，如果缓存决策决定在内容回传时存储内容块，需要执行缓存替换算法。为了将更高缓存收益率的内容块留在缓存中，在最近最少使用 (LRU, least recently used) 算法的基础上，结合内容块的访问代价  $c_n^k$  进行缓存替换。记内容块  $k$  最近一次访问时刻与当前的时间间隔为  $t_k$ ，则替换时选择  $\frac{t_k}{c_n^k}$  最大的内容块移出缓存。

将  $\frac{1}{c_n^k}$  看作权值，不妨将该替换算法记为加权 LRU 算法。

### 4.3 复杂度分析

本小节分别分析缓存决策算法的空间复杂度和时间复杂度。

#### 1) 空间复杂度

虽然一个内容块并没有在布隆过滤器中，但是由于散列冲突的存在，布隆过滤器可能会返回错误的结果，为满足错误率要求，需要保证布隆过滤器的长度不能低于某一下限。记布隆过滤器的比特长度为  $l$ ，容纳内容块的个数为  $w$ ，给定错误率  $f$ ，布隆过滤器的长度可以由式(15)得到<sup>[25]</sup>

$$l = -\frac{w \ln f}{(\ln 2)^2} \quad (15)$$

其中， $w$  为一个时隙内到达的内容块个数，由于同一个内容块在一个时隙内被请求 2 次的概率较小，可以近似得到

$$w \approx \lambda_n s_i = \lambda_n \frac{1}{g_n} \quad (16)$$

其中， $\lambda_n$  为节点  $n$  的内容请求到达率，假定相邻的时隙中节点缓存内容的命中收益变化不大，根据式

(13)，有  $\tilde{g}_n \approx \tilde{g}_n(s_i) = \frac{\eta_n \tilde{c}_n \lambda_n}{v_n}$ ，其中， $\eta_n = \frac{\sum_{k \in K_n} h_{s_i}^k}{\lambda_n s_i}$  为

节点  $n$  在时隙  $s_i$  内的缓存命中概率， $\tilde{c}_n = \frac{\sum_{k \in K_n} c_n^k h_{s_i}^k}{\sum_{k \in K_n} h_{s_i}^k}$  为

缓存命中内容平均访问代价，代入式(15)得

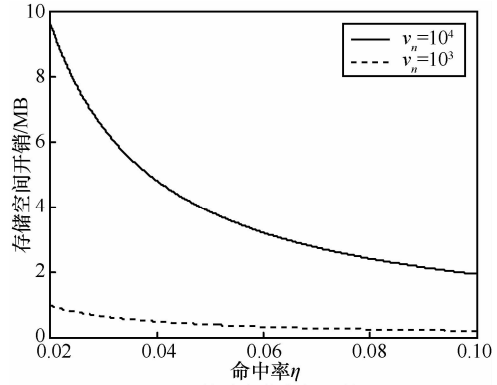
$$l \approx -\frac{v_n \ln f}{\tilde{\eta} \tilde{c}_n (\ln 2)^2} \quad (17)$$

若内容块的最大访问代价为  $c_{\max}$ ，则算法运行需要存储  $2c_{\max} + 1$  个布隆过滤器，式(17)说明算法空间开销与内容请求速率无关。图 5 (a) 给出

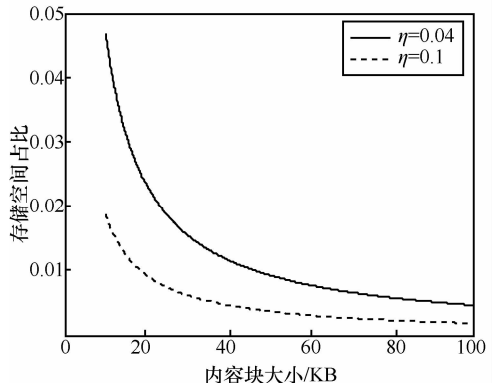
了  $f = 10^{-8}$ ， $c_{\max} = 10$ ， $\tilde{c}_n = 5$  时决策算法的存储空间开销，相比文献 [20] 内容块的总数为  $1.083 \times 10^7$  时 32.47~32.64 MB 的存储空间开销，具有较大的优势。

记  $u$  为单个内容块的大小，则占用空间与缓存空间的比值为

$$\mu = \frac{(2c_{\max} + 1)l}{v_n u} = -\frac{(2c_{\max} + 1) \ln f}{\tilde{\eta} \tilde{c}_n u (\ln 2)^2} \quad (18)$$



(a) 算法存储空间开销



(b) 算法存储空间占比

图 5 算法空间复杂度分析 ( $f = 10^{-8}$ ， $c_{\max} = 10$ ， $\tilde{c}_n = 5$ )

式(18)表明，当错误率  $f$  和节点拓扑位置给定时，算法存储空间开销占比与主要与缓存命中概率  $\eta$  和单个内容块的大小有关。当  $f = 10^{-8}$ ， $c_{\max} = 10$ ， $\tilde{c}_n = 5$  时， $\mu$  值随内容块大小  $u$  的变化如图 5 (b) 所示，可以看出该算法在能够在存储总空间占用小于 5% 的情况下支持大于 10 KB 粒度的数据块的缓存决策。

#### 2) 时间复杂度

决策算法的计算开销主要包括散列函数计算开销和布隆过滤器查询开销。按照散列函数的最优个数  $d = \frac{l \ln 2}{w}$ ，根据式(15)得



$$d = -\frac{\ln f}{\ln 2} \quad (19)$$

设定每个时隙对应的布隆过滤器长度相同，则各个布隆过滤器使用同一组散列函数即可。当  $f=10^{-8}$  时，对  $d$  向上取整，得到每当有一个请求到达时，该机制需要计算的散列次数为 27，与文献 [20] 中的最好情况相同。

为加速对布隆过滤器查询匹配过程，对  $2c_{\max} + 1$  个布隆过滤器的存储位置进行合并，即各个布隆过滤器位置相同的比特位放置在存储器的连续空间中，从而查询时通过一次存储器访问即可读取多个布隆过滤器的比特位，大幅减少由于存储器访问带来的时间开销。

另外，决策机制执行在内容请求到达时，但并不影响请求消息的转发，从而请求转发和缓存决策算法可以并行执行，最大限度地减少缓存决策带来的处理时延。

## 5 仿真与性能分析

### 5.1 仿真参数设置

采用 ICN 网络仿真工具 ndnSIM<sup>[26,27]</sup> 进行仿真与性能分析。网络拓扑采用 Rocketfeul 项目测量到的真实网络拓扑 (sprint PoP level topology)<sup>[28]</sup>，该拓扑包含 52 个节点和 84 条链路，图 6 给出了该拓扑的可视化显示。网络中的内容块总数为  $10^8$ ，大小相同，均为 50 KB，链路带宽均为为 1 Gbit/s。在网络中部署 4 个内容源服务器，各服务器存储  $2.5 \times 10^7$  个内容块，并随机选取 4 个节点与内容源服务器直接连接，并根据最短路径路由由算法建立转发表。选取所有连接度为 1 的 19 个节点作为用户接入节点，每个节点的用户内容请求到达服从参数  $\lambda=100$  个/秒的泊松过程，请求概率服从 Zipf 分布。

为了构造内容请求的动态变化特征，在内容请求对应 Zipf 分布序列中随机选取一定比例的扰动内容，进行随机化重新排序<sup>[29]</sup>，扰动比例  $q$  选取 0~25%。参数  $\alpha$  的取值范围为 0.6~1.3，节点缓存容量与内容文件总空间比值  $C$  的变化范围为 0.01%~0.09%，即节点能够缓存  $1 \times 10^4 \sim 9 \times 10^4$  个内容块。

初始状态缓存中的内容块是完全随机分布的，为获取稳定状态下的性能指标，在系统运行 2 000 s 后再开始数据统计，统计时长为 3 000 s。

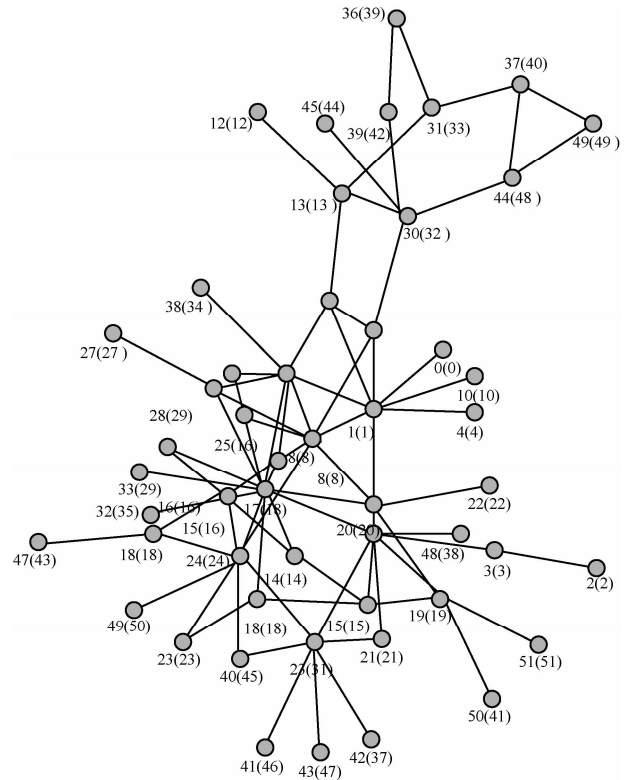


图 6 仿真拓扑结构

### 5.2 性能指标

将本文提出的 GACS 与 CE<sup>[23]</sup>，ProbCache+<sup>[10]</sup> 和 MAGIC<sup>[16]</sup> 方案进行对比分析，CE<sup>2</sup> 和 ProbCache+ 分别使用 LRU 算法和加权 LRU 算法进行缓存替换，MAGIC 通过统计最近 500 s 内的内容请求到达次数获取内容流行度信息。性能评价指标包括：路由跳数减小率 (HRR, hop reduction ratio)、缓存命中率 (CHR, cache hit ratio) 和缓存替换数 (CRN, cache replace number)。

#### 1) 路由跳数减小率

定义路由跳数减小率为相比请求消息在源服务器端获取响应，由于缓存命中，传输的路由跳数减少的程度

$$HRR = 1 - \frac{\sum_{i \in R} h_i}{\sum_{i \in R} H_i} \quad (20)$$

其中， $h_i$  为请求  $i$  在缓存网络中经过的路由跳数， $H_i$  为请求  $i$  在源服务器端获取响应传输的路由跳数， $R$  为请求消息集合。该指标反映缓存系统带宽节省的能力。

#### 2) 缓存命中率

定义缓存命中率为网络中节点缓存内容响应

请求的概率

$$CHR = 1 - \frac{\omega}{|R|} \quad (21)$$

其中， $\omega$  为在源服务器端获取响应的请求消息数量，该指标反映缓存系统降低服务器负载的能力。

### 3) 缓存替换数

定义缓存替换数为平均每个节点缓存替换的内容块数量

$$CRN = \frac{\sum_{n \in N} r_n}{|N|} \quad (22)$$

其中， $r_n$  为节点  $n$  发生的缓存替换次数。该指标反映缓存替换带来的存储器操作维护开销，同时间接反映 GACS 对高缓存收益内容的误判情况。

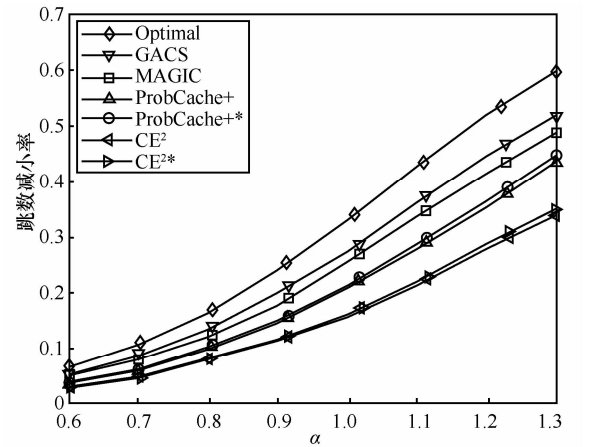
## 5.3 仿真结果分析

下面分别分析各个变量不同取值下的性能指标仿真结果。另外，为更好地分析 GACS 性能，按照缓存收益优化模型式(1)~式(4)，使用贪婪算法，计算出在缓存协作的情况下，理论最优解以及对应的 HRR 和 CHR，记理论最优解为 Optimal。

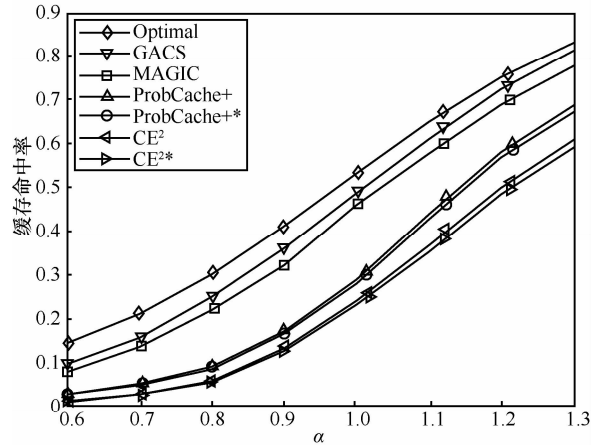
### 1) 性能指标随参数 $\alpha$ 的变化情况

图 7 分别给出了缓存容量占比  $C=0.05\%$ ，扰动比例为  $q=10\%$  时，4 种方案的各个性能指标随 Zipf 分布参数  $\alpha$  的变化情况。其中，使用加权 LRU 算法的  $CE^2$  和 ProbCache+ 记为  $CE^{2*}$  和 ProbCache+\*，用虚线表示，且 CRN 指标只与决策算法有关，不再区分替换算法的不同。由于  $CE^2$  缓存所有经过节点的内容块，节点的同质化缓存将会导致内容缺失概率增大，对应的 HRR 和 CHR 最小，且 CRN 最大；ProbCache+ 是一种概率缓存机制，不再缓存所有的内容块，有效降低了 CRN，以更大的概率在远离内容源服务器的节点决策缓存，提高了内容在距离请求者更近的节点被缓存的概率，在一定程度上提高了缓存内容的多样性，但该方案缺乏对内容请求特征的精细化考量，HRR 和 CHR 增幅有限。在一定程度上提高了缓存内容的多样性，但该方案缺乏对内容请求特征的精细化考量，HRR 和 CHR 增幅有限。MAGIC 在传输路径上选择缓存收益增幅最大的节点存储内容，但是高收益内容块可能需要经过多次请求后才会被复制到网络边缘。而 GACS 令节点直接缓存具有更高收益的内容块，相比 MAGIC，

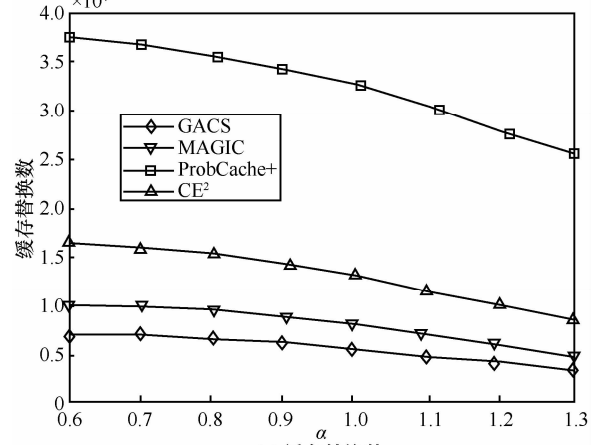
网络边缘节点更快速地缓存高收益内容，再加上 MAGIC 默认采用固定间隔的访问次数统计方法，对流行度的变化敏感度不高，所以 GACS 在 HRR 和 CHR 方面表现更优。由于 GACS 对缓存收益优化模型进行了近似，而且缓存机制在运行过程中需要逐渐发现流行度的扰动，而理论最优解对内容流行度变化的反应是瞬时的，从而 GACS 的 HRR



(a) 路由跳数减小率



(b) 缓存命中率



(c) 缓存替换数

图 7 性能指标随参数  $\alpha$  的变化 ( $C=0.05\%$ ,  $q=10\%$ )

和 *CHR* 与理论最优解存在一定的差距。高收益内容有一定的误判率，在缓存替换数方面比 *MAGIC* 稍多。另外，加权 *LRU* 算法在缺少匹配的决策算法的情况下，难以大幅优化内容块的放置情况，对 *CE<sup>2</sup>* 和 *ProbCache+* 的 *HRR* 提升程度非常有限，同时由于权值的作用，*CHR* 略有降低。

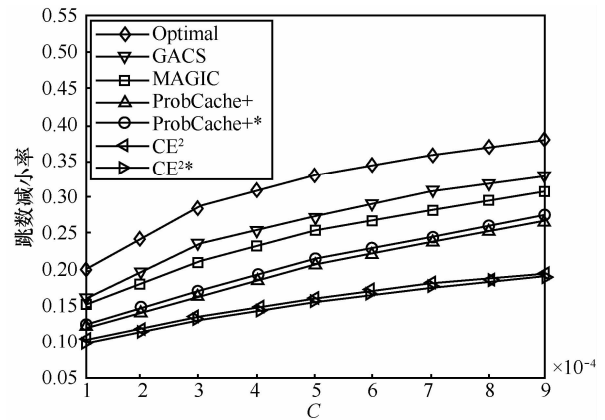
在变化趋势方面，随着参数  $\alpha$  的增大，内容流行度分布的集中性不断加强，流行内容块拥有更大的请求概率，从而 4 种方案的 *HRR* 和 *CHR* 不断增大。由于缓存替换发生在缓存未命中时，所以 4 种方案随参数  $\alpha$  和 *CHR* 的增大，*CRN* 不断减小。而且根据 4.1 节的分析，*GACS* 的误判率随着参数  $\alpha$  的增大而降低，*GACS* 与 *MAGIC* 的 *CRN* 差距逐渐缩小。

2) 性能指标随缓存容量占比 *C* 的变化情况

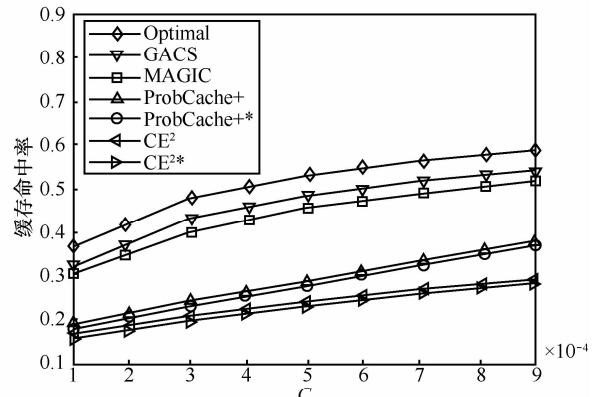
图 8 分别给出了 Zipf 分布参数  $\alpha=1$ ，扰动比例为  $q=10\%$  时，4 种方案的各个性能指标随缓存容量占比 *C* 的变化情况。随着缓存容量的增大，缓存系统能够存储更多的内容块，4 种方案的 *HRR* 和 *CHR* 也随之提高，其中，*GACS* 仍然具有最高的 *HRR* 和 *CHR*，但是由于增加的缓存空间存储的内容流行度相对较低，增幅逐渐趋缓。在 *CRN* 方面，*CE<sup>2</sup>* 由于缓存命中率的提升，*CRN* 随缓存容量的增加而减小，由于 *ProbCache+* 根据缓存容量确定缓存概率，其 *CRN* 随缓存容量的增加而增大，而 *MAGIC* 和 *GACS* 由于缓存决策主要与内容流行度的变化和 内容请求速率有关，与缓存容量的大小无关，所以缓存替换数变化不大。

3) 性能指标随扰动比例的变化情况

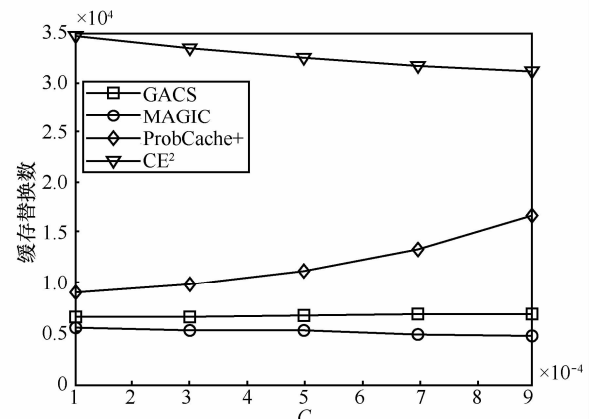
图 9 分别给出了 Zipf 分布参数  $\alpha=1$ ，缓存容量占比  $C=0.05\%$  时，4 种方案的各个性能指标随扰动比例  $q$  的变化情况，除了理论最优解直接根据扰动的流行度计算得到，其他各方案都需要一定的时间适应流行度的变化，扰动比例的提高对 4 种方案的性能表现都产生了不利影响。由于 *MAGIC* 根据一段时间内的平均请求到达率判断内容流行度的高低，在内容流行度分布稳定时其性能指标最优，当扰动比例提高时，*GACS* 对内容流行度的跟踪能力得到体现，*HRR* 和 *CHR* 开始优于 *MAGIC*，且扰动比例越高，优势越大，在 *CRN* 方面，随着扰动比例的提高和 *CHR* 的降低，*MAGIC* 的 *CRN* 增幅较快，*GACS* 与 *MAGIC* 的差距逐渐缩小。



(a) 路由跳数减小率



(b) 缓存命中率



(c) 缓存替换数

图 8 性能指标随缓存容量占比 *C* 的变化 ( $\alpha=1, q=10\%$ )

根据上述仿真结果综合分析，由于 *CE<sup>2</sup>* 缓存所有经过节点的内容块，容易造成缓存冗余，缓存系统的整体效用不高，且缓存替换频繁，存在较多的缓存操作开销。*ProbCache+* 采用加权概率的方式，缓存内容的概率与节点和请求者之间的距离成反比，在大幅降低缓存替换频率的同时获取了一定的性能提升，但缓存决策时仍然存在一定的随机性和盲目性。*MAGIC* 根据精确的内容块访问到达率和节点与内容源的距离进行缓存决策，进一步提升了缓存的针对性，但是

对内容流行度变化的敏感度不高，且在内容块数量庞大的 ICN 缓存系统中存在较大的存储和计算开销。GACS 利用缓存对内容请求流的过滤效应，基于存储开销较小的流行度感知机制，准确地捕获缓存收益较高的内容块，较好地节省了内容访问总代价，并以有限的误判率为代价，提高了对内容流行度变化的跟踪能力，在内容流行度时变的实际网络环境中，综合性能更优。另外，MAGIC 需要收集沿途节点的内容块访问频率信息，而 GACS 不需要节点间交互，在部署应用方面更为灵活。

## 6 结束语

本文针对 ICN 缓存系统细粒度内容流行度获取问题，建立了以最大化降低内容访问代价为目标的缓存收益优化模型，提出了一种基于收益感知的缓存机制。该机制利用上游缓存对内容请求流的过滤效应，节点独立地完成缓存决策的同时达到一定的缓存协作效果；将内容流行度获取和缓存决策过程融合考虑，根据从源服务器获取内容的代价和访问频率计算缓存收益，使用基于布隆过滤器的滑动窗口方法检测访问到达间隔，并在缓存替换方面使用加权 LRU 算法，实现了高缓存收益内容的捕获；根据缓存内容平均收益调整时隙长度，动态适配请求速率变化，以较低的存储和计算开销实现了基于内容块流行度的缓存决策，仿真结果和性能分析显示了其有效性。后续研究工作包括：1) 分析不同路由机制对 GACS 的影响，研究设计与之匹配的缓存路由机制；2) 验证不同网络拓扑和带宽条件下 GACS 的性能。

## 参考文献：

- [1] XYLOMENOS G, VERVERIDIS C N, SIRIS V A, et al. A survey of information-centric networking research[J]. Communications Surveys & Tutorials, 2014, 16(2): 1024-1049.
- [2] 张国强, 李杨, 林涛, 等. 信息中心网络中的内置缓存技术研究[J]. 软件学报. 2014, 25(1): 154-175.
- [3] ZHANG G Q, LI Y, LIN T, et al. Survey of in-network caching techniques in information-centric networks[J]. Journal of Software, 2014, 25(1):154-175.
- [4] JACOBSON V, SMETTERS D K, THORNTON J D, et al. Networking named content[J]. Communications of the ACM, 2012, 55(1): 117-124.
- [5] BORST S, GUPTA V, WALID A. Distributed caching algorithms for content distribution networks[C]//The IEEE INFOCOM. San Diego, CA, c2010: 1-9.
- [6] SAINO L, PSARAS I, PAVLOU G. Hashing routing schemes for information-centric networking[C]//The 3rd ACM SIGCOMM Workshop on Information-Centric Networking. New York, USA, c2013: 27-32.
- [7] 葛国栋, 郭云飞, 刘彩霞, 等. 命名数据网络中基于局部请求相似性的协作缓存路由机制[J]. 电子与信息学报, 2015, 37(2): 435-442.
- [8] GE G D, GUO Y F, LIU C X, et al. Collaborative caching and routing scheme based on local request similarity in named data networking[J]. Journal of Electronics & Information Technology, 2015, 37(2): 435-442.
- [9] LAOUTARIS N, SYNTILA S, STAVRAKAKIS I. Meta algorithms for hierarchical Web caches[C]//2004 IEEE International Conference on Performance, Computing, and Communications. Phoenix, USA, c2004: 445-452.
- [10] EUM S, NAKAUCHI K, MURATA M, et al. CATT: potential based

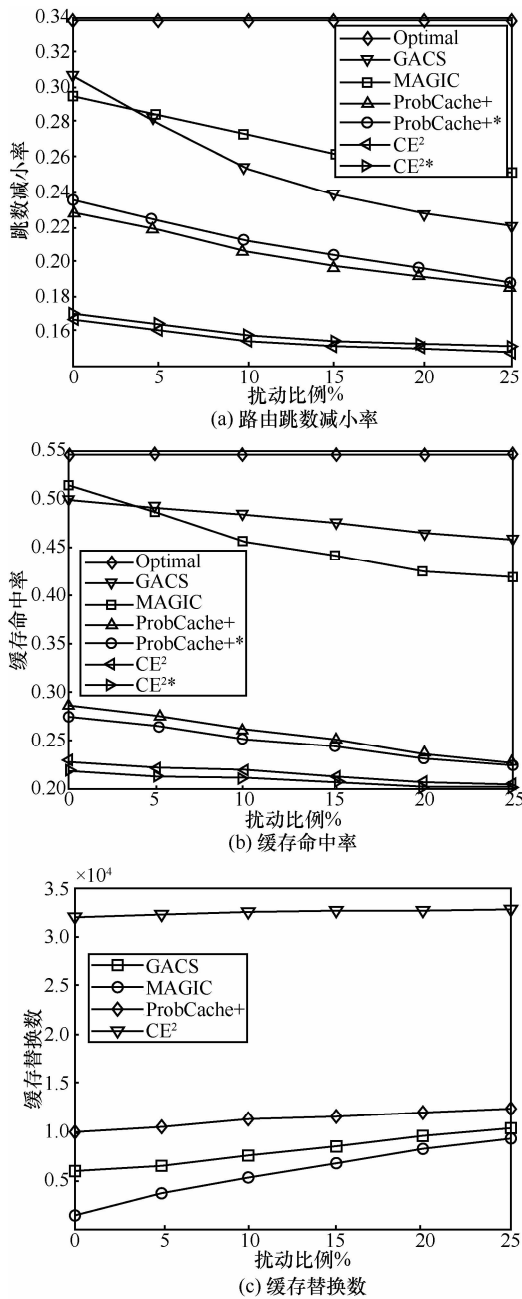


图 9 性能指标随扰动比例的变化 ( $\alpha=1, C=0.05\%$ )

- routing with content caching for ICN[C]//ACM SIGCOMM Workshop on Information-Centric Networking. Helsinki, Finland, c2012: 49-54.
- [9] PSARAS I, CHAI W K, PAVLOU G. Probabilistic in-network caching for information-centric networks[C]//ACM SIGCOMM Workshop on Information-Centric Networking. Helsinki, Finland, c2012: 55-60.
- [10] PSARAS I, WEI K C, PAVLOU G. In-network cache management and resource allocation for information-centric networks[J]. Parallel and Distributed Systems, IEEE Transactions on, 2014, 25(11): 2920-2931.
- [11] KIDEOK C, MUNYOUNG L, KUNWOO P, et al. WAVE: popularity-based and collaborative in-network caching for content-oriented networks[C]//2012 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). Orlando, FL, c2012: 316-321.
- [12] CHAI W, HE D, PSARAS I, et al. Cache less for more in information-centric networks(extended version)[J]. Computer Communications, 2013, 36(7): 758-770.
- [13] 蔡君, 余顺争, 刘外喜. 基于节点社团重要度的 ICN 缓存策略[J]. 通信学报, 2015, 36(6): 2015222.  
CAI J, YU S Z, LIU W X. Caching strategy based on node's importance to community in information-centric networks[J]. Journal on Communications, 2015, 36(6): 2015222.
- [14] MING Z X, XU M W, WANG D. Age-based cooperative caching in information-centric networks[C]//2012 IEEE INFOCOM Workshop on Emerging Design Choices in Name-Oriented Networking. Orlando, USA, c2012: 268-273.
- [15] IOANNOU A, WEBER S. Towards on-path caching alternatives in information-centric networks[C]//2014 IEEE 39th Conference on Local Computer Networks (LCN). Edmonton, AB, c2014: 362-365.
- [16] 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.
- [17] JING R, WEN Q, WESTPHAL C, et al. MAGIC: a distributed max-gain in-network caching strategy in information-centric networks[C]. 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). Toronto, ON, c2014: 470-475.
- [18] BADOV M, SEETHARAM A, KUROSE J, et al. Congestion-aware caching and search in information-centric networks[C]//The 1st International Conference on Information-Centric Networking. Paris, France, c2014: 37-46.
- [19] MELAZZI N B, BIANCHI G, CAPONI A, et al. A general, tractable and accurate model for a cascade of LRU caches[J]. Communications Letters, IEEE, 2014, 18(5): 877-880.
- [20] DAI H, WANG Y, WU H, et al. Towards line-speed and accurate on-line popularity monitoring on NDN routers[C]//2014 IEEE 22nd International Symposium of Quality of Service (IWQoS). Hong Kong, China, c2014: 178-187.
- [21] KIM Y, YEOM I. Performance analysis of in-network caching for content-centric networking[J]. Computer Networks, 2013, 57(13): 2465-2482.
- [22] 朱轶, 糜正琨, 王文鼎. 一种基于内容流行度的内容中心网络缓存概率置换策略[J]. 电子与信息学报, 2013, 35(06): 1305-1310.
- ZHU Y, MI Z K, WANG W N. A cache probability replacement policy based on content popularity in content centric networks[J]. Journal of Electronics & Information Technology, 2013, 35(6): 1305-1310.
- [23] YE H E, HO T, CUI Y, et al. VIP: a framework for joint dynamic forwarding and caching in named data networks[C]//The 1st International Conference on Information-Centric Networking. Paris, France, c2014: 117-126.
- [24] BRESLAU L, CAO P, FAN L, et al. Web caching and Zipf like distributions: evidence and Implications[C]//IEEE International Conference on Computer Communications (INFOCOM). New York, USA, c1999: 126-134.
- [25] TARKOMA S, ROTHENBERG C E, LAGERSPETZ E. Theory and practice of Bloom filters for distributed systems[J]. IEEE Communications Surveys & Tutorials, 2012, 14(1): 131-155.
- [26] AFANASYEV A, MOISEENKO I, ZHANG L X. ndnSIM: NDN simulator for NS-3[R]. NDN, Technical Report NDN-0005, 2012.
- [27] NS-3 based named data networking(NDN) simulator[EB/OL]. <http://ndnsim.net>, 2014.
- [28] SPRING N, MAHAJAN R, WETHERALL D, et al. Measuring ISP topologies with rocketfuel[J]. IEEE/ACM Transactions on Networking, 2004, 12(1): 2-16.
- [29] GUO S, XIE H Y, SHI G. Collaborative forwarding and caching in content centric networks[C]//IFIP Networking. Prague, Czech Republic, c2012: 41-55.

#### 作者简介:



陈龙 (1988-), 男, 山东济宁人, 国家数字交换系统工程技术研究中心博士生, 主要研究方向为信息中心网络、移动互联网。

汤红波 (1968-), 男, 湖北孝感人, 国家数字交换系统工程技术研究中心教授、硕士生导师, 主要研究方向为新型网络体系结构、移动通信网络。

罗兴国 (1951-), 男, 重庆人, 国家数字交换系统工程技术研究中心教授、博士生导师, 主要研究方向为移动通信网络、高性能计算机体系结构。

柏溢 (1975-), 男, 江苏盐城人, 国家数字交换系统工程技术研究中心讲师, 主要研究方向为信息中心网络、移动通信网络。

张震 (1985-), 男, 山东济宁人, 国家数字交换系统工程技术研究中心讲师, 主要研究方向为新型网络体系结构、信息中心网络。