



## 一种基于 SDN 的 CCN 集中控制缓存决策方法

曲桦<sup>1</sup>, 李岩松<sup>1</sup>, 刘军<sup>1</sup>, 赵季红<sup>1,2</sup>, 郭涯<sup>1</sup>

(1. 西安交通大学, 陕西 西安 710049; 2. 西安邮电大学, 陕西 西安 710061)

**摘要:** 在软件定义网络 (SDN) 和内容中心网络 (CCN) 融合架构下, 为了充分利用控制层对网络拓扑和缓存资源的全局感知, 在全网中实现缓存资源的优化使用, 提出了一种集中控制的缓存决策优化方案。在该方案中, 应用粒子群优化算法 (PSO) 并且根据节点边缘度、节点重要度以及内容流行度对缓存资源和内容进行集中缓存决策, 使得内容在不同的节点进行合理的缓存。仿真结果表明, 通过评估缓存大小对缓存性能的影响, PSO 缓存决策方法取得了比 LCE、PROB 缓存决策策略更优的缓存命中率和路径延展率, 明显降低了缓存节点的缓存替换数, 使得缓存达到了整体缓存优化。

**关键词:** 软件定义网络; 内容中心网络; 缓存; 粒子群优化; 节点边缘度; 节点重要度; 内容流行度

**中图分类号:** TN915.9

**文献标识码:** A

**doi:** 10.11959/j.issn.1000-0801.2017132

## A method of CCN centralized control cache decision based on SDN

QU Hua<sup>1</sup>, LI Yansong<sup>1</sup>, LIU Jun<sup>1</sup>, ZHAO Jihong<sup>1,2</sup>, GUO Ya<sup>1</sup>

1. Xi'an Jiaotong University, Xi'an 710049, China

2. Xi'an University of Posts and Telecommunications, Xi'an 710061, China

**Abstract:** In the software defined networking and content centric networking integration framework, a cache optimization scheme of centralized control was proposed to make full use of global awareness of the control layer for network topology and cache resources, and achieve the optimal use of caching resources in the entire network. In this scheme, the particle swarm optimization (PSO) was applied for centralized cache decision-making depending on node edge degree, node importance degree and content popularity. Therefore, the contents were cached in different nodes reasonably. Simulation result shows that the proposed PSO caching scheme achieves larger cache hit rate and lower path stretch than that of LCE scheme, PROB scheme by evaluating the impact of cache size and content popularity on the caching performance. It also shows that PSO scheme significantly reduces the number of cache nodes to replace the cache, making the cache to achieve the overall cache optimization.

**Key words:** software defined networking, content centric networking, caching, particle swarm optimization, node edge degree, node importance degree, content popularity

收稿日期: 2017-01-17; 修回日期: 2017-05-07

基金项目: 国家自然科学基金资助项目 (No.61371087, No.61531013); 教育部—中国移动科研基金资助项目 (No.MCM20150102)

**Foundation Items:** The National Natural Science Foundation of China (No.61371087, No.61531013), Joint Research Foundation of Ministry of Education and China Mobile (No.MCM20150102)

## 1 引言

随着社交网络、云存储、视频网站和数据类应用等的流行,网络呈现出以信息内容为中心的发展趋势。根据思科虚拟网络指数的预测,到2018年各种形式的多媒体流量会达到网络流量的90%<sup>[1]</sup>,网络的使用已经偏离了现有网络当初的设计原则和设想,现有网络在安全性、移动性、可靠性、可用性、可扩展性等方面存在着极大的挑战<sup>[2]</sup>。

面对上述问题,内容中心网络(content centric networking, CCN)<sup>[3]</sup>应运而生,CCN是一种能够满足用户信息传递需求的新型体系架构,它采用以内容为中心的通信方式。区别于传统网络主机到主机的通信,用户不再关注内容的位置,而只关注内容本身,传输模式由传统通信中的“推”改为适合内容海量传输的“拉”,提高了内容获取的传输效率和网络内容资源的利用率。在CCN架构中,网络内置缓存是其核心的思想之一,引起了研究者们的广泛关注。

目前的缓存策略分为路径缓存和邻域缓存<sup>[4,5]</sup>,缓存决策使用的信息为传输路径上或邻域上节点和内容的信息,虽然提高了缓存的命中率和路径延展率,但是这些信息来源于传输路径或邻域等局部网络,具有局限性。CCN中的缓存呈现泛在化、网络化的新特征,缓存的拓扑结构需用任意图的网状结构来描述,增加了数学建模和分析的难度,也使缓存协同变得更为困难<sup>[6]</sup>,节点在缓存决策时独立做出决策,因此不能很好地提高缓存效率。CCN中的缓存是工作在网络层的串联设备,要求以线速执行,因此要求缓存管理策略简单,极大地降低了缓存的有效性。

将软件定义网络(software defined networking, SDN)<sup>[7]</sup>的概念加入CCN中,能够有效地解决CCN中的缓存问题。SDN是一种当下热门的网络架构,在SDN中将网络控制和数据转发进行

解耦,数据层由网络基础设备组成,仅提供简单的数据转发功能,可以快速处理匹配的数据分组;控制层具有逻辑中心化和可编程的控制器,掌握全局网络信息。两层之间采用开放的统一接口进行交互。控制器通过标准接口向网络基础设备下发统一标准规则,网络基础设备仅需按照这些规则执行相应的动作即可。因此SDN具有动态管理和编程控制网络的能力。将SDN和CCN进行融合,既可以发挥SDN逻辑中心化和集中控制的优势,又可以发挥CCN在内容获取方面的优势。最近,研究者们发表了很多关于SDN和CCN融合的科研成果<sup>[8-10]</sup>,在SDN和CCN的融合架构下,缓存决策有了更加高效和智能的方法。粒子群优化(particle swarm optimization, PSO)算法<sup>[11]</sup>是近年来发展起来的一种新的群智能优化算法,是通过群体中个体之间的协作和信息共享来寻找最优解,粒子群算法简单、容易实现,并且没有过多参数需要调整。

本文在SDN和CCN融合架构下,提出了一种基于SDN的CCN下的缓存决策方法,利用控制器具有逻辑中心化以及对全网感知的优势,根据全局网络拓扑和网络内容的信息,对内容和缓存进行集中和整体的优化;控制器周期性地对缓存信息进行统计,并在接收到数据层缓存请求后进行缓存决策;本方法还将节点的重要度和边缘度以及内容的流行度对节点和内容进行整体的数学建模,并应用粒子群优化算法进行优化。仿真结果表明,通过控制层对内容的集中缓存决策,使得内容在不同的节点进行合理的缓存,相比于传统网络的缓存方法,PSO缓存决策方法明显地提高了缓存命中率、降低了路径延展率,并且能够极大地降低缓存替换数。

## 2 相关研究

内置缓存技术是信息中心网络(information centric networking, ICN)的关键技术,它充分利



用网内缓存资源, 节约网内带宽, 改善整体网络性能。CCN 最初采用的处处缓存 (leave copy everywhere, LCE) 机制<sup>[3]</sup>, 将请求内容缓存到沿途传输的所有节点中, 该方法会产生大量的冗余副本, 对缓存空间的利用率不高。

针对传统缓存机制的弊端, Bernardini 等人<sup>[12]</sup>将突破点放在内容的流行度上, 提出了一种基于内容流行度 (most popular content, MPC) 的缓存管理策略, 即节点只缓存流行度高的内容。这可以减少每个节点的缓存负载, 实现了高缓存命中率, 提高了网络资源的利用率。但是该方案也没有考虑内容流行度的实时性问题。参考文献[4]提出了一种基于加权概率的缓存机制, 返回路径中沿途节点缓存内容的概率与节点和用户的距离成反比, 此方法提高了内容在用户附近节点缓存的概率, 但是也增加了不同内容在边缘缓存处的竞争。参考文献[5]提出了基于中介中心度的缓存策略, 用户请求通过中介中心度大的节点的概率比较高, 通过在中介中心度高的节点处缓存内容可以提高缓存命中率, 但内容在中介中心度大的节点处的替换率较高, 反而降低了缓存命中率。

黄胜等人<sup>[13]</sup>将节点分为两类: “边缘”类节点和“核心”类节点, 提出了一种基于节点分类 (based node classification, BNC) 的数据存储策略, 对不同类型的节点按照不同的存储策略进行存储, 并考虑用户请求数据时的实时性。数据的缓存位置由节点位置与数据在节点中的流行度共同决定, 适合层次化结构网络。当多用户同时请求内容时, 不同的下载速度会降低缓存命中率, 为此, Sasano 等人<sup>[14]</sup>在流量控制下对多用户的内容下载进行了仿真, 发现通过延长节点上内容的缓存时间, 可以提高 CCN 中多用户同时请求内容的下载完成时间, 但是它只在含有两个用户的简单拓扑中实现。

引入“协作”概念, Wang 等人<sup>[15]</sup>提出了一种自治系统域内协作的缓存模式, 在自治系统内进行问题的构建, 消除邻域节点中的缓存冗余, 并

提出当用户请求未得到响应时, 向邻域节点发出请求。这种方法充分利用了网内资源, 但向邻域节点周期性地发送通告以及 AS 内缓存冗余的消除, 则需要额外的系统开销。SDN 架构具有全局视野, 通过集中控制对网络的逻辑控制功能和高层策略灵活地进行动态、优化配置, 进而实现网络资源的优化配置。借助 SDN 的优势, 雷方元等人<sup>[16]</sup>提出了一种基于 SDN 的 ICN 高效缓存机制, 将网络中的缓存节点根据社团重要度划分成重要度不一的社团, 利用 SDN 控制面感知全网状态, SDN 控制器根据感知网络社团的重要度和社团网络节点的重要度来取定缓存位置策略, 并在此基础上生成概率替换策略。在这种方法中, 重要度较高的社团负载比较大, 同时也存在较大的缓存后路由代价。

基于传统缓存机制的弊端, 研究者通过寻找不同的创新点, 突破原始的缓存框架, 借鉴了不同的解决方法, 引入了新颖的概念思路, 从沿途缓存到旁路缓存, 从非协作模式到协作模式, 从分布式独立运行到集中控制, 缓存机制的性能得到很大的提高, 网络的整体性能也得到了充分的发挥。本文在 SDN 和 CCN 融合架构下, 提出了一种在路径域上对内容进行集中缓存决策的方法, 通过 SDN 控制层对全局的感知和集中控制, 将内容合理地缓存在不同节点上, 能够有效地提高网络中缓存的利用率。

### 3 基于 SDN 的 CCN 缓存决策方法

#### 3.1 相关定义

**定义 1** (节点重要度) 交换机在接收到控制器下发的流表时, 对下发的流表信息进行统计, 根据交换机统计的信息可以得到每个内容的路由路径经过此交换机的次数, 次数越多则节点对于该内容的重要度越大。用  $B_{ik}$  表示节点  $i (i=1, \dots, n)$  对于内容  $k (k=1, \dots, m)$  的重要度, 并将其归一化, 可表示为  $b_{ik} = B_{ik} / B_{\max}^i$ ,  $B_{\max}^i$  表示节点  $i$  对于内容  $k$

的最大节点重要度。节点重要度能够真实地反映出不同缓存节点对于内容的请求次数,次数越多,节点对于该内容的重要度越大。

**定义 2** (节点边缘度) 控制器进行缓存决策时应用了节点边缘度的概念,节点离用户越近,边缘度越大。将内容缓存在节点边缘度较大的节点,则用户终端请求该节点内容时的时延更短,能够为用户提供更好的用户体验。节点  $i$  的边缘度定义为  $L_i$ ,表示节点距离内容服务器的跳数。节点  $i$  的边缘度归一化可表示为  $l_i = L_i / L_{\max}$ ,  $L_{\max}$  表示所有节点中最大的边缘度。

**定义 3** (内容流行度) 每个节点通过对内容请求分组的统计,得到每个内容的请求次数,将节点在一个周期内的内容请求次数作为内容在该节点处的内容流行度。内容流行度越大,说明内容在该节点处将来被请求的概率越大。用  $P_{ik}$  表示内容  $k$  在节点  $i$  处的流行度,其归一化为  $p_{ik} = P_{ik} / P_{\max}^i$ ,  $P_{\max}^i$  表示内容在节点  $i$  处的最大流行度。

网络内置缓存的功能由网络内的节点承载,根据现有网络研究,网络呈幂律分布<sup>[17]</sup>,因此,节点在网络拓扑中处于不同位置,导致了节点在网络中的重要程度不同,而节点的重要程度决定了缓存的价值,当节点处于重要位置时,缓存的价值就更大,内容缓存在该位置被命中的概率就

更大。本文中的节点重要度不仅考虑节点在网络中的位置,还考虑了内容的请求路径。节点边缘度反映了网络拓扑中节点离用户的远近程度,内容倾向于缓存距离用户比较近的节点。通过节点重要度和节点边缘度甄别出价值更大的缓存。内容流行度则体现了每个节点对于不同内容的需求,内容流行度越高,则说明内容在此节点的需求就越大。本文中,SDN 控制器通过感知全网拓扑信息,计算节点重要度、节点边缘度以及内容流行度等参数信息,并将其转化为 PSO 模型的输入,经过该算法决策后,输出为一个或多个内容的最优缓存节点。所以,PSO 和 SDN 之间存在着很强的依赖关系。

### 3.2 缓存决策机制

基于 SDN 的 CCN 缓存决策实施流程如图 1 所示,控制层中的控制器负责数据层缓存信息的收集以及缓存决策,数据层中的交换机负责对信息的统计以及执行控制器缓存决策后的缓存动作。控制层和数据层之间通过 OpenFlow 协议进行通信。SDN 控制器和 CCN 节点的通信通过 OpenFlow 协议进行交互,采用 CCN 方式,通过流表下发告知 CCN 交换节点缓存决策结果。

数据层中的交换机实时地对缓存信息进行统计,交换机中包含以下两张表:节点重要度表,记录各个内容经过此交换机的路由路径数量,主

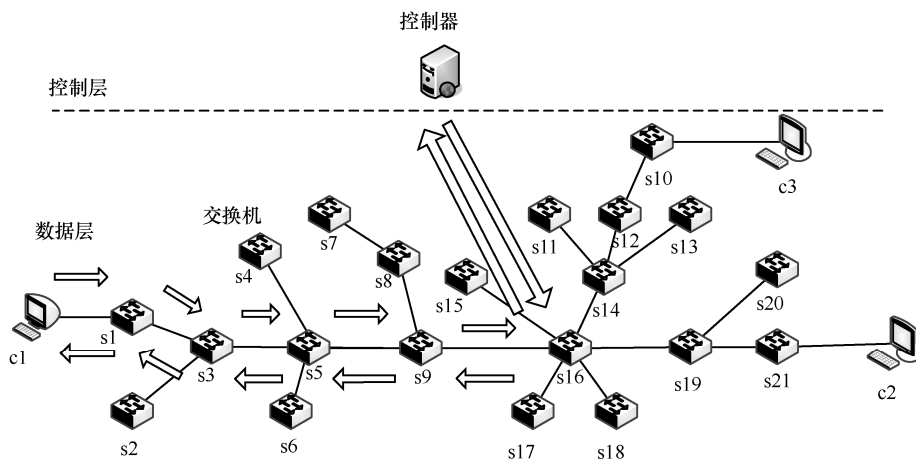


图 1 基于 SDN 的 CCN 缓存决策实施流程



要包括内容名和路由路径数量两项；内容流行度表，记录各个内容的请求次数，包括内容名和请求次数两项。除了收集信息外，交换机还负责执行缓存决策后缓存的动作，当数据层的交换机接收到内容分组后，首先对内容分组进行判断，若内容分组已标记为已记录缓存决策结果，则从内容分组中读取缓存决策结果，并判断缓存结果是否有自身节点，若有，则缓存该内容，若没有，则继续传输该内容分组；若内容分组未标记，说明该内容分组中没有缓存决策结果，则交换机向控制器发送包含该内容名的缓存决策请求。例如，当用户 *c1* 发出内容请求后，请求沿着 *s1*→*s3*→*s5*→*s9*→*s16* 路径进行传输，最后在交换机 *s16* 处命中，*s16* 沿着原路径发出内容分组，因为是首次请求，该内容分组未标记，所以 *s16* 会向控制层发出请求决策消息，控制器通过粒子群优化算法计算后，通过流表下发，将缓存决策结果发送给 *s16*，*s16* 随后将该内容标记为已决策，并将缓存结果封装到该内容分组中，处理完毕后，将内容分组发出，沿原路径返回，*s16*、*s9*、*s5*、*s3*、*s1* 依次读取该内容分组中的缓存结果，若缓存结果中包含自身节点，则缓存该内容并继续传输，反之则不缓存，只需继续传输该内容分组即可，最终将该内容返回给用户 *c1*。

控制层中的控制器周期性地收集缓存信息节点重要度和内容流行度，并根据网络拓扑模块提供的信息计算每个节点的节点边缘度。当控制器接收到数据层的缓存决策请求后，首先判断相关内容是否有缓存结果，若没有，则根据缓存信息进行粒子群优化算法，得到结果后将结果下发给提出缓存决策请求的交换机；若有，则直接将结果下发给交换机。

### 3.3 缓存决策目标模型

通过节点重要度和节点边缘度将网络拓扑中的节点分为重要节点和非重要节点，通过内容流行度也将内容进行区分，通过将不同流行度的内

容合理地缓存在不同重要程度的节点中，可以明显地优化节点缓存。其优化过程充分利用 SDN 和 CCN 融合架构下逻辑中心化的优势，对节点和内容应用粒子群优化算法进行集中的优化处理，将不同流行度的内容合理地缓存在不同重要程度的节点。其优化单位为内容的传输路径，对路径上的节点和内容进行集中缓存决策。

根据节点重要度和节点边缘度以及内容流行度对缓存决策进行数学建模。节点重要度高的节点代表内容的路由路径数量多，内容请求经过此节点的概率比其他节点大，边缘节点为节点边缘度比较高的节点，缓存在此节点的内容被请求时可以提供较低的时延。将流行度较高的内容缓存在节点重要度和节点边缘度高的节点，将流行度较低的内容缓存在节点重要度和节点边缘度低的节点，能够明显地优化缓存决策。根据以上分析得到将内容 *k* 缓存在节点 *i* 的优化函数为：

$$f_{ik} = \alpha e^{c_1 \|l_i - p_{ik}\|} + \beta e^{c_2 \|b_k - p_{ik}\|} \quad (1)$$

其中， $\alpha$ 、 $\beta$  分别代表缓存节点时节点边缘度和节点重要度的权重，且  $\alpha + \beta = 1$ ， $c_1$ 、 $c_2$  为常数。以路径为优化单位构建优化问题如下：

$$\min F = \sum_{i \in D} \sum_{k \in C_i} f_{ik} \quad (2)$$

其中，*D* 表示路径上节点的集合，*C<sub>i</sub>* 表示节点 *i* 上缓存内容的集合。

### 3.4 缓存决策算法

具有代表性的 CCN 缓存决策算法有 LCE 算法、PROB 算法，在用户内容请求初期，能够表现出较快的响应速度，但随着用户内容请求的增加，带来较高的缓存替换率，会造成较高的时延。而 PSO 可以同时多个内容进行缓存决策，将多个内容合理地缓存到缓存节点上，提高了缓存效率，虽然缓存决策过程需要一定的决策时间，但从整体上来看，PSO 能够满足用户内容请求的实时性要求。

控制器接收到缓存决策请求后，根据收集到

的缓存信息以及内容传输路径上的节点, 采用 PSO 进行缓存决策优化过程, PSO 应用的具体过程如下。

#### (1) 定义编码方式

缓存策略的优化变量是内容类, 对路径上节点统计的内容按照内容流行度进行排序, 并且根据排序对内容进行分类, 每个类中包含一定数量的流行度相近的内容。PSO 根据适应度函数将路径上的节点和分类后的内容集进行匹配, PSO 中的每个粒子包括 3 个组成部分: 位置、速度和适应度值。针对第  $s$  个粒子, 采用整数编码, 其编码形式为:

$$X_s = \{x_{s1}, \dots, x_{sw}\}, V_s = \{v_{s1}, \dots, v_{sw}\} \quad (3)$$

其中,  $x_{si} (i=1, \dots, w)$  表示分配给路径中第  $i$  个节点的某类内容集合编号,  $v_{si} (i=1, \dots, w)$  表示分配给路径中第  $i$  个节点的内容集合编号的更新速度。适应度值是该粒子由适应函数得到的优化结果。

#### (2) 初始化粒子群

随机初始化所有粒子的位置和速度, 根据粒子的最大速度, 在速度域值内随机选择一个值作为粒子的初始化速度, 根据内容集合编号, 在内容集合编号值域内随机选择值作为粒子的初始化位置, 将其作为每个粒子的最优解  $pbest_{si}$ , 通过搜索粒子群得到全局最优解  $gbest_i$ 。

#### (3) 位置、速度更新策略

根据粒子群的最优解和每个粒子的局部最优解更新每个粒子的速度, 通过粒子的速度对粒子群中的每个粒子进行位置的更新, 从而进入下代粒子群。粒子的速度和位置更新公式为:

$$v_{si}(t+1) = wv_{si}(t) + c_1r_1(pbest_{si}(t) - x_{si}(t)) + c_2r_2(gbest_i(t) - x_{si}(t)) \quad (4)$$

$$x_{si}(t+1) = x_{si}(t) + v_{si}(t+1) \quad (5)$$

其中,  $t$  和  $t+1$  表示迭代次数,  $w$  表示保持原来速度的系数,  $c_1$ 、 $c_2$  分别表示粒子跟踪个体最优值和粒子群全局最优值的权重,  $r_1$  和  $r_2$  是  $[0,1]$

内均匀分布的随机数。

#### (4) 解码策略和收敛性检查

由于标准 PSO 适用于求解连续解空间问题, 可以将粒子由连续非整型变量转化为离散整型变量, 方法是将非整数解码为最相近的整数值。由于 PSO 收敛速度较快, 判断收敛的方法为: 可以预先定义最大迭代次数来判断; 判断全局最优解在特定的迭代次数之内没有变化则收敛。

### 4 实验仿真与结果分析

#### 4.1 仿真环境和性能评价指标

本文使用 ndnSIM 仿真工具, 并用 C++ 编程实现了 SDN 和 CCN 融合架构的内容传输模型以及 CCN 内容传输模型。为了验证 PSO 缓存策略方法是否达到预期效果, 选择 CCN 中具有代表性的 LCE、PROB 两种缓存策略作为对比。主要仿真参数设置见表 1。

表 1 主要仿真参数设置

| 参数              | 设置值 |
|-----------------|-----|
| 网络模型            | B-A |
| 节点个数/个          | 21  |
| 用户数/个           | 3   |
| 服务器个数/个         | 1   |
| 内容数/条           | 100 |
| 用户内容请求次数/次      | 800 |
| 内容热门度模型 Zipf 参数 | 0.8 |
| 内容请求到达泊松过程参数    | 25  |
| 缓存替换算法          | LRU |

尽管 CCN 支持多径路由传输, 本文仅在单径路由前提下对缓存策略进行性能分析, 并采取以下测量参数作为性能评价指标。

#### (1) 缓存命中率

缓存命中率 (CHR) 越高, 表示缓存响应用户缓存请求的概率就越大, 从而提升缓存系统效率, 改善网络负载。CHR 可表示为:

$$CHR = \frac{n}{N} \quad (6)$$



其中,  $n$  表示用户内容请求在节点处得到响应的个数,  $N$  表示用户内容请求的总数。

## (2) 路径延展

路径延展 (PS) 反映了用户请求得到响应的节点距离用户的远近, PS 越低, 表示响应用户请求的节点离用户越近, 这不仅能够明显改善用户的体验, 而且能有效降低网络负载。PS 可表示为:

$$PS = \frac{d}{P} \quad (7)$$

其中,  $d$  表示仿真中用户发出的请求获取内容的平均跳数,  $P$  表示用户的内容请求从源服务器得到响应的平均跳数。

## (3) 缓存替换数

通常情况下, 网内缓存容量有限, 远远小于内容的大小, 这势必会产生大量的替换操作。替换操作不仅会消耗计算资源, 还会导致缓存副本在节点的存活时间太短, 从而影响缓存命中率。因此减少缓存命中率 (CRN) 也是一个需要考虑的问题。

## 4.2 仿真结果

当缓存大小变化时, 观察 LCE、PROB、PSO 3 种缓存决策策略的缓存命中率、路径延展和缓存替换数的变化情况。

缓存命中率随节点缓存大小的变化情况如图 2 所示, 其中横坐标表示缓存与内容的比率, 即每个节点的缓存大小除以内容数。假设内容集合为 1~100 共 100 个, 缓存大小为 1 (1 个缓存单元缓存 1 个内容), 则缓存与内容的比率为 1%。由图 2 可知, 随着节点缓存的不断增大, 3 种机制的缓存命中率不断提高, 而 PSO 取得了明显高于其他 2 种机制的缓存命中率。路径延展率随节点缓存大小的变化情况如图 3 所示, 由图 3 可知, 随着节点缓存的不断增大, 3 种机制的路径延展率不断降低, PSO 的路径延展率取得了更好的结果。与传统的 2 种缓存策略机制相比, PSO 对路径域上的节点和内容进行集中缓存决策, 不仅实现了节点之间的有效协同, 还对内容在节点上进行了更加

合理的布局, 因此在缓存命中率和路径延展率方面取得了更好的效果。

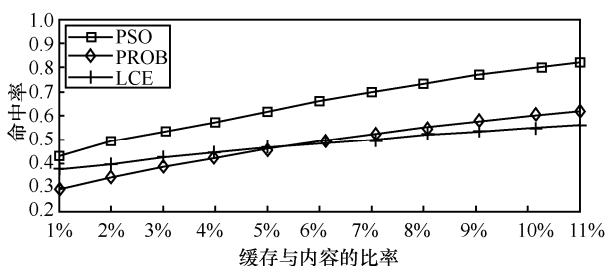


图 2 缓存命中率随节点缓存大小的变化情况

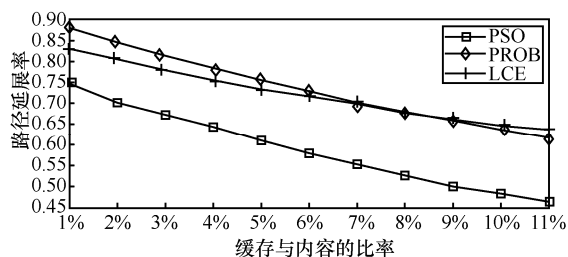
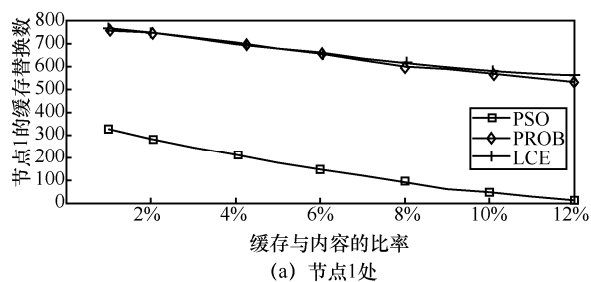
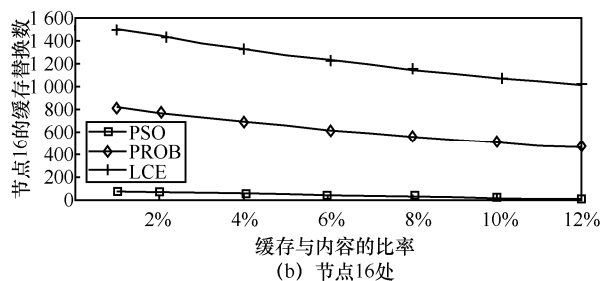


图 3 路径延展率随节点缓存大小的变化情况

缓存替换数随节点缓存大小的变化情况如图 4 所示。由图 4 可知, 当节点缓存不断增大时, 3 种缓存机制的缓存替换数不断减小, 而 PSO 相比其他 2 种缓存决策机制取得了更稳定和更优的结果。PSO 缓存决策方法周期性地根据目标模型对缓存进行集中决策, 能够合理地将内容缓存在相应的节点上, 有效地减少了内容的替换。



(a) 节点1处



(b) 节点16处

图 4 缓存替换数随节点缓存大小的变化情况

## 5 结束语

本文在 SDN 和 CCN 融合架构下, 提出了一种利用 SDN 框架全局感知和集中控制优势的 PSO 缓存决策方法, 并在该方法中引入节点重要度、节点边缘度以及内容流行度的概念, 通过问题构建对路径域上的内容进行集中的缓存决策, 同时考虑了节点之间的协同和内容在节点上的合理分布。SDN 和 CCN 结合的网络架构, 是未来网络研究的一个方向, 要实现和现网的融合, 可以将 CCN 的通信模式改为传统 IP 方式, 这个问题的解决可以采用散列的方法<sup>[18]</sup>。此外, 仿真结果表明, PSO 缓存决策方法随着节点缓存大小的变化都表现出了比其他 2 种缓存决策策略更好的缓存性能。虽然本文提出的缓存决策方法充分考虑了内容感知和集中控制的优势, 对路径域内进行了集中优化, 但是仍然属于局部的整体优化, 在全局内整体进行缓存优化则能取得更好的缓存性能。下一步将考虑在全局范围内利用路由和缓存进行联合优化。

## 参考文献:

- [1] INDEX V N. Cisco visual networking index: forecast and methodology, 2010-2015[J]. White Paper Cisco Systems Inc, 2011: 1-16.
- [2] FELDMANN A. Internet clean-slate design: what and why?[J]. Acm Sigcomm Computer Communication Review, 2007, 37(3): 59-64.
- [3] JACOBSON V, SMETTERS D K, THORNTON J D, et al. Networking named content[C]// International Conference on Emerging Networking Experiments and Technologies, December 1-4, 2009, Rome, Italy. New York: ACM Press, 2009: 117-124.
- [4] PSARAS I, CHAI W K, PAVLOU G. Probabilistic in-network caching for information-centric networks[C]// Edition of the ICN Workshop on Information-Centric Networking, August 13-17, 2012, Helsinki, Finland. New York: ACM Press, 2012:55-60.
- [5] WEI K C, DILIANG H, IOANNIS P, et al. Cache "less for more" in information-centric networks[M]. Berlin: Springer Berlin Heidelberg, 2012.
- [6] ZHANG G Q, YANG L I, TAO L, et al. Survey of in-network caching techniques in information-centric networks[J]. Journal of Software, 2014, 25(1): 154-175.
- [7] ONF. Software-defined networking: the new norm for networks[S/OL]. (2012-04-13)[2017-01-17]. [http://bigswitchnetworks.com/sites/default/files/sdn\\_resources/onf-whitepaper.pdf](http://bigswitchnetworks.com/sites/default/files/sdn_resources/onf-whitepaper.pdf).
- [8] EUM S, JIBIKI M, MURATA M, et al. A design of an ICN architecture within the framework of SDN[C]// International Conference on Ubiquitous and Future Networks, July 7-10, 2015, Sapporo, Japan. New Jersey: IEEE Press, 2015: 141-146.
- [9] VELTRI L, MORABITO G, SALSANO S, et al. Supporting information-centric functionality in software defined networks[C]// IEEE International Conference on Communications, June 10-15, 2012, Ottawa, Canada. New Jersey: IEEE Press, 2012: 6645-6650.
- [10] SUN Q, WANG W, HU Y, et al. SDN-based autonomic CCN traffic management[C]// GLOBECOM Workshops, December 8-12, 2014, Austin, USA. New Jersey: IEEE Press, 2015: 183-187.
- [11] 杨维, 李歧强. 粒子群优化算法综述[J]. 中国工程科学, 2004, 6(5): 87-94.  
YANG W, LI Q Q. Survey on particle swarm optimization algorithm[J]. Engineering Sciences, 2004, 6(5): 87-94.
- [12] BERNARDINI C, SILVERSTON T, FESTOR O. Cache management strategy for CCN based on content popularity[M]. Berlin: Springer Berlin Heidelberg, 2013.
- [13] 黄胜, 滕明埏, 吴震, 等. 命名数据网络中一种基于节点分类的数据存储策略[J]. 计算机研究与发展, 2016(6): 1281-1291.  
HUANG S, TENG M N, Wu Z, et al. A data caching scheme based on node classification in named data networking[J]. Journal of Computer Research and Development, 2016(6): 1281-1291.
- [14] SASANO K, BANDAI M, YAMAMOTO M. A cache management method for simultaneous downloading from multiple receivers for content centric networking[C]// International Conference on Advanced Information Networking and Applications Workshops, March 23-25, 2016, Crans-Montana, Switzerland. New Jersey: IEEE Press, 2016: 872-877.
- [15] WANG J M, ZHANG J, BENSOU B. Intra-AS cooperative caching for content-centric networks[C]// ACM Sigcomm Workshop on Information-Centric Networking, August 12-16, Hong Kong, China. New York: ACM Press, 2013: 61-66.
- [16] 雷方元, 蔡君, 罗建桢, 等. 一种基于 SDN 的 ICN 高效缓存机制[J]. 计算机科学, 2016, 43(8): 74-78.  
LEI F Y, CAI J, LUO J Z, et al. Efficient caching mechanism based on soft defined information-centric networks[J]. Computer Science, 2016, 43(8): 74-78.
- [17] BARABASI A L, ALBERT R. Emergence of scaling in random networks[J]. Science, 1999, 286(5439): 509.
- [18] OOKA A, ATA S, KOIDE T, et al. OpenFlow-based content-centric networking architecture and router implementa-