

基于节点中心性度量的内容中心网络缓存机制

蔡岳平^{1,2}, 刘军², 樊欣唯²

(1. 重庆大学信息物理社会可信服务计算教育部重点实验室, 重庆 400030;

2. 重庆大学通信工程学院, 重庆 400030)

摘要: 为了降低内容中心网络的缓存内容冗余度和提高缓存内容命中率, 提出一种基于节点中心性度量的缓存机制(CMC)。CMC 利用控制器获取整个网络的拓扑结构和缓存空间空闲率, 根据拓扑的连接关系分别计算各节点的度中心性、紧密中心性和介数中心性, 并将这 3 个度量和缓存空间空闲率作为缓存节点选择的重要指标。仿真实验表明, 与传统内容中心网络 (CCN) 缓存机制相比, CMC 可以有效地提高缓存内容命中率, 降低获取内容的跳数和平均请求时延。

关键词: 内容中心网络; 缓存机制; 中心性; 度量

中图分类号: TP393

文献标识码: A

Node centrality metric based caching mechanism in content-centric network

CAI Yue-ping^{1,2}, LIU Jun², FAN Xin-wei²

(1. Key Laboratory of Dependable Service Computing in Cyber Physical Society, Chongqing University, Chongqing 400030, China;

2. College of Communication Engineering, Chongqing University, Chongqing 400030, China)

Abstract: In order to reduce the cache redundancy as well as increase the cache hit ratios in content-centric networks, the node centrality metric based caching mechanism (CMC) was proposed. CMC utilized controllers to obtain the topology of the whole network and the idle rate of cache space. According to the connection relation of the topology, the degree centrality, closeness centrality and betweenness centrality of nodes were calculated. When CMC choose the caching nodes, it took the three metrics and the idle rate of cache space into account. Simulation results show that CMC can effectively increase the cache hit ratios and reduce the content fetching hops and average request delay compared with the traditional routing algorithms in CCN.

Key words: content-centric network, caching mechanism, centrality, metric

1 引言

当前, 基于 TCP/IP 协议的互联网产生于二十世纪六七十年代, 解决了当时主机端到端的互联互通问题, 实现了硬件资源的共享。随着互联网应用的快速发展, 特别是物联网、云计算和移动互联网等应用的发展, 接入互联网的设备数量大

量增加, 互联网应用的主体已经转向文字信息和多媒体音视频等内容的获取, 信息内容服务已经成为网络服务的主体, 用户更关注内容的获取, 而非内容存放的地址或主机。传统的基于 TCP/IP 协议的互联网体系结构难以满足未来互联网应用的发展。在网络的可扩展性、移动性、安全性、可控可管性、位置依赖性以及拥塞控制和路由策

收稿日期: 2016-12-02; 修回日期: 2017-05-08

基金项目: 国家自然科学基金资助项目 (No.61301119); 教育部高等学校博士学科点专项科研基金资助项目 (No.20120191120025); 教育部留学归国人员启动基金资助项目 (No.1020607820140002)

Foundation Items: The National Natural Science Foundation of China (No.61301119), Research Fund of Young Scholars for the Doctoral Program of Higher Education, Ministry of Education (No.20120191120025), The Scientific Research Foundation for the Returned Overseas Chinese Scholars, Ministry of Education (No.1020607820140002)

略等方面都表现出诸多的不适应性。为了从根本上解决这些问题，研究者们提出了以 DONA^[1]、PURSUIT^[2]、NetInf^[3]和 CCN (content centric network)^[4]等为代表的信息中心网络 (ICN, information centric network)。ICN 以信息为中心，其关注点从传统网络的网络位置转向信息本身，网络中的各个节点和程序在各种信息请求和应答行为的驱动下运行，而 ICN 的功能就是协调相关命名数据的传输和缓存。其中内容中心网络 (CCN) 被认为是最有前途的方案之一，其核心是通过对内容资源的直接命名和基于内容名称的路由进行内容的分发和获取，其网络节点除了具有传统网络节点所具有的路由和转发能力外，还具备存储内容资源及服务内容请求的功能。

CCN 有 2 种分组格式：兴趣分组和数据分组。兴趣分组是请求者发出的内容请求分组；数据分组是内容服务节点（缓存节点或源服务器）通过兴趣分组的反向路径将内容信息传输给请求者的内容分组。CCN 的节点承担着数据分组的存储、转发和路由任务。CCN 节点维持 3 类信息表：内容存储表 (CS, content store)、待定兴趣表 (PIT, pending interest table) 和转发信息表 (FIB, forwarding information base)。CS 的作用是缓存流经该节点的内容资源并为后续内容请求提供服务。PIT 用于记录请求信息的到达端口，从而建立数据分组的反向路径，同时具有聚合相同内容请求的作用。FIB 用来将数据分组发送到目的端，可以同时向多个方向转发请求。当兴趣分组到达路由节点后，首先查询 CS 中是否有内容 A 的缓存资源。若有，直接将内容 A 返回给请求者 R，并丢弃兴趣分组；若没有，则查询 PIT 中是否有内容 A 的请求记录。若有，则在相应条目增加该兴趣分组的输入端口并丢弃兴趣分组；若没有，则在 PIT 中新增条目并查询 FIB 获取兴趣分组的下一跳信息。当数据分组按照 PIT 建立的反向路径将数据分组传输给请求者时，各个路由节点会按照一定的缓存策略对数据进行缓存。

缓存机制^[5]是内容中心网络的核心技术之一。其核心是通过在网络内的路由器上缓存内容资源，使内容请求能够使用最近的缓存副本而无需通过寻址找到主机后再获取相应内容，提高内容传输效率，降低网络中的冗余流量以及缓解服务器的访问压力等，从而提高网络的性能。缓存机

制设计对 CCN 的内容分发效率具有重要影响，特别是缓存位置的选择，合理的缓存位置选择能使用户更易于从邻近缓存节点获取内容资源。

LCE (leave copy everywhere)^[4]是 CCN 早期的缓存机制，它要求数据分组转发路径上的所有节点，无差别地对数据分组中的内容资源进行缓存。该方案会产生大量的缓存冗余，缓存效率较低。LCD (leave copy down)^[6]使内容对象只缓存其所在节点的下一跳节点，缓存内容到达边缘节点速度慢，且会产生大量缓存冗余。MCD (move copy down)^[6]在缓存命中时将缓存内容从命中节点向下游移动一位（源服务器除外），可以减少缓存冗余，但当请求者来自不同的边缘网络时，会出现内容缓存点的摇摆，这种动态性会产生更多的网络开销。Prob^[6]要求数据分组返回路径上的所有路由节点都以固定的概率缓存对象。ProbCache^[7]中请求的对象根据概率存放在每个节点中，但概率都不同，概率与请求节点的距离成反比。文献[8]提出一种端系统间相互协作的 CONIC 机制，缓存内容放置在边缘用户处，路由节点指引请求信息前往临近用户获取内容资源，但该方案中的路由器仅用于路径指引，未缓存内容。文献[9]提出一种基于内容生存期的缓存机制，只有内容的生存期超时内容才被替换，而内容的生存期与内容的缓存位置和流行度有关，即内容越接近网络边缘，内容流行度越高，内容的生存期越长。文献[10,11]提出了基于介数的缓存机制，该机制将内容对象只缓存在路径介数最大的路由节点上。文献[12,13]将软件定义网络 (SDN, software defined network)^[14]应用到内容中心网络的缓存内容管理中。

本文提出了一种基于节点中心性度量的缓存机制 (CMC, centrality metric based caching)，利用控制器获取整个网络的拓扑结构，根据拓扑的连接关系分别计算各节点的相关度量，并据此得到若干个缓存节点，从而实现在网络开销较低的情况下达到缓存效率最优的目的。

本文的贡献如下。

1) 选取 3 个节点中心性度量作为网络拓扑中节点重要性的评价指标，并对 3 个度量进行综合处理，量化各节点的重要性。

2) 提出一种基于节点中心性度量的内容中心网络缓存机制 (CMC)，用于降低缓存内容冗余度并提高缓存效率。

2 节点中心性度量概述

本节将对本文所使用的节点中心性度量进行介绍。复杂网络是在结构、节点多样性和连接多样性等方面呈现高度复杂性的网络，一个复杂网络可以通过一系列包含系统结构信息的度量指标分析网络拓扑结构属性。网络中节点的价值首先取决于该节点在网络中所处的位置，位置越趋近中心的节点分析价值就越大。中心性是复杂网络分析的核心内容之一，它可用于发现网络中最具重要意义的节点。

复杂网络可以用无向图 $G=(V, E)$ 表示，假设 G 有 n 个节点， m 条边，用 $V=\{v_1, v_2, \dots, v_n\}$ 表示 G 的顶点集合， $E=\{e_1, e_2, \dots, e_m\}$ 表示 G 的边集合。图 G 的邻接矩阵 $A=(a_{ij})_{n \times n}$ ，其中，当 v_i 与 v_j 有边相连时， $a_{ij}=1$ ；否则， $a_{ij}=0$ 。

2.1 度中心性

度通常被看作获取网络上流动内容的直接程度。度中心性 (degree centrality) 是网络分析中刻画节点中心性的最直接度量指标。它衡量节点的邻居个数。一个节点的度越大就意味着这个节点的度中心性越高，该节点越趋于网络的枢纽节点。对于一个 n 个节点的无向图 $G=(V, E)$ ，节点 v_i 的度中心性和节点序列号 i 与其他 $n-1$ 个节点之间的直接联系的数量成正比，其计算式为

$$C_D(v_i) = \frac{\sum_{j=1}^n a_{ij}}{n-1} \quad (1)$$

其中， $C_D(v_i)$ 表示节点 v_i 的度中心性，为了消除网络规模对度中心性的影响，将节点 v_i 的度数除以其他 $n-1$ 个节点最大可能的连接数，得到与节点 v_i 有直接相连的网络节点的比例，所以，度中心性的取值范围为 $0 \sim 1$ 。

2.2 紧密中心性

紧密中心性 (closeness centrality) 反映在网络中某一节点与其他节点之间的紧密程度。紧密中心性是由某节点到网络中其他节点最短距离来确定的，越高的紧密中心性意味着这个节点距离网络中其他节点越接近，也就越趋近网络的中心。在无向图中，其计算式为

$$C_C(v_i) = \sum_{t \in V \setminus \{v_i\}} 2^{-d_G(v_i, t)} \quad (2)$$

其中， $C_C(v_i)$ 表示节点 v_i 的紧密中心性， $d_G(v_i, t)$ 表示 G 中节点 v_i 到节点 t 的最短路径长度。在无向图

中，将 $C_C(v_i)$ 除以 $n-1$ 进行归一化。

2.3 介数中心性

介数刻画的是节点控制信息传输的能力。介数中心性 (betweenness centrality) 是衡量节点对信息在整个网络内传播的重要性的度量。它是由节点出现在其他节点之间的最短路径上的情况来决定的，通常具有高介数中心性的节点是网络中的关键节点。对于一个拥有 n 个节点的无向图 $G=(V, E)$ ，节点 v_i 的介数中心性的计算步骤如下。

1) 计算每对节点 (s, t) 之间所有的最短路径。

2) 对于每对节点 (s, t) ，判断节点 v_i 是否处于它的最短路径上，用包含节点 v_i 的最短路径条数占两点间最短路径总条数的比例来表示介数中心性。

3) 对每对节点 (s, t) 求出的结果进行累加，得出的结果就是节点 v_i 的介数中心性。其计算式为

$$C_B(v_i) = \sum_{s \neq v_i, t \in V} \frac{\sigma_{st}(v_i)}{\sigma_{st}} \quad (3)$$

其中， $C_B(v_i)$ 表示节点 v_i 的介数中心性， σ_{st} 表示节点对 (s, t) 之间所有的路径数， $\sigma_{st}(v_i)$ 表示节点对 (s, t) 之间经过节点 v_i 的路径数。在无向图中，将 $C_B(v_i)$ 除以 $\frac{n(n-1)}{2}$ 进行归一化。

尽管上述 3 个中心性度量均能衡量节点的重要度，但是不同的中心性度量在各个方面或多或少都存在着一一定的不足和局限性。虽然度中心性的计算复杂度最低、最易理解，但是它仅考虑了节点的局部信息，未考虑网络的全局结构。当有 2 个节点分别属于不同的连通片，那么用紧密中心性度量节点重要度就无法得到合理的结果，这使紧密中心性在与网络断开连接的连通片中缺乏适用性。介数中心性虽然考虑了网络的全局结构，并且能用于断连的连通片网络，但是其仍然有一定的不足。例如，在真实网络中有很大部分节点都不出现于任何 2 个节点之间的最短路径上，从而使这些大部分节点的介数中心性值都为 0。由于不同的中心性度量的机制不同，而且有着不同的缺陷，因此，当对同一个网络使用不同的中心性度量进行节点重要度评估时，往往会得到不同的结果。所以本文考虑融合这 3 种节点中心性度量来对节点重要度进行评估。文献[15]将度中心性、紧密中心性和介数中心性这 3 个最为基础的节点中心性度量作为多个属性引入到 TOPSIS 多属性决策模型中，将融合后的节点中心性度量作为网络节点的重要度评估标准。该算法

不仅解决了这3个中心性各自存在的缺陷，还有效地将它们的差异进行折中融合。而本文采用更为简单的加权叠加的方法对3个基础的节点中心性度量进行融合，并用融合后的度量值作为缓存节点选择的标准。

3 基于节点中心性度量的缓存机制 CMC

假定CCN由内容提供者、内容请求者和具有缓存功能的路由节点组成。CCN中包含兴趣分组和数据分组2种分组格式。路由节点可以缓存内容，同时维护和查询转发信息表(FIB)、待定兴趣表(PIT)和内容存储表(CS)。内容以分块的形式进行传输且各个内容块大小相同。内容请求者以发送兴趣分组的方式向网络的内容提供者或缓存节点请求相应内容。

3.1 CMC的工作机制

CMC的核心思想是基于节点中心性度量理论，将内容缓存位置选在网络拓扑中的关键节点上，这些节点相对于其他节点具有较高的度中心性、紧密中心性和介数中心性。由于机制运行时需要知道整个网络的拓扑结构，而SDN控制器具有全局的视角，所以将软件定义网络控制器用于拓扑信息收集及处理，下面详细介绍CMC的工作机制。

1) 兴趣分组解析。假设一个请求者R想要获取一个内容名称为A的内容资源，并向网络发出兴趣分组。该兴趣分组首先会在内容路由器的内容存储表中查询是否有内容A的缓存资源。若有，直接将内容A返回给请求者R，并丢弃兴趣分组；若没有，则查询PIT中是否有内容A的请求记录。若有，则在相应条目增加该兴趣分组的输入端口并丢弃兴趣分组；若没有，则在PIT中新增条目并查询FIB获取兴趣分组的下一跳信息。按此流程兴趣分组被逐跳转发至内容提供者P。

在此过程中，控制器收集网络拓扑信息及链路时延信息，在原始的紧密中心性和介数中心性的计算过程中，以最小跳数来衡量最短路径。但由于链路负载与网络拥塞的存在，跳数并不能很好地反映网络的状态，当跳数较小，但链路负载较大，网络发生拥塞时，链路时延可能较大，对网络性能的影响也较大。链路时延能更好地反映网络状态，所以本文在计算紧密中心性 $C_C(v_i)$ 和介数中心性 $C_B(v_i)$ 时，是以最小路径时延来衡量最短路径。分组在网络中的端到端时延具体可分为4个部分：传播时延 T_{trans} 、发送时延 T_{send} 、处理时延 T_{proc} 和排队时延

T_{queue} 。 T_{trans} 为分组在链路上的传播时延，与链路的物理传输介质和距离有关。 T_{send} 表示分组在路由节点的发送时延，与路由节点端口速率和分组大小有关。分组在路由节点中的总时延 T_{node} 等于处理时延 T_{proc} 与排队时延 T_{queue} 之和。网络分组端到端时延分析模型可由式(4)表示。

$$T_{delay} = \sum_{i=1}^H T_{trans}(i) + \sum_{j=1}^N (T_{send}(j) + T_{queue}(j) + T_{proc}(j)) \quad (4)$$

其中， H 为分组路由路径的链路跳数， N 是分组路由路径中的路由节点数。假定分组在路由节点中处理和排队过程服从M/G/1排队过程。通过式(4)就可以计算出路径的分组端到端时延，再以最小路径时延来衡量最短路径，并依此计算紧密中心性和介数中心性。然后对3个中心性度量进行归一化处理，并依据不同的需求确定3个度量所占权重分别为 α 、 β 和 γ ，并计算得出各个内容路由器的度量总分 $S(v_i)$ ，其计算式为

$$S(v_i) = \alpha C_D(v_i) + \beta C_C(v_i) + \gamma C_B(v_i) \quad (5)$$

计算出各内容路由器的度量总分后，控制器依据度量总分 $S(v_i)$ 和缓存空间空闲率 $I(v_i)$ 计算得到兴趣分组转发路径上各个内容路由器的缓存优先级 $P(v_i)$ ，其计算式为

$$P(v_i) = I(v_i) \frac{S(v_i)}{\sum_{i=1}^h S(v_i)} \quad (6)$$

其中， h 为兴趣分组转发路径上的内容路由器的个数。计算出兴趣分组转发路径上各个内容路由器的缓存优先级后，控制器会对这些内容路由器的缓存优先级值进行排序，并依据业务需求按照优先级值排序依次选取若干个内容路由器作为缓存节点，向其发送主动缓存内容的指令。

2) 数据分组转发。当兴趣分组到达内容提供者P后，内容提供者P会发出一个包含内容A的数据分组，并依据PIT中的记录沿着兴趣分组的反向路径进行转发。携带内容A的数据分组除了会被转发至请求者R外，还会执行主动缓存内容A的指令将内容A缓存在控制器选出的内容路由器上。

3) 缓存注销。当缓存空间不足时，为了缓存新到达的内容资源，内容路由器需要使用缓存替换算法LRU(least recently used)^[16]或LFU(least frequently used)^[16]替换已缓存的内容资源。

3.2 CMC的工作实例

下面，结合具体实例对CMC机制进行说明。

图 1 是实例说明使用的网络拓扑 NSFNET^[17], NSFNET 是美国国家科学基金会 (NSF) 为了满足各大学及政府机构研究工作的迫切要求, 在全美国建立的连接各大超级计算中心的网络, 包含 14 个网络节点和 21 条链路。将节点 College PK MD 作为请求者的连接节点, 请求者发出的兴趣分组将会首先到达该节点; 将节点 Salt Lake City UT 作为服务者的连接节点, 当提供者发布内容时, 内容注册信息将会首先到达该节点; 其余节点为普通的路由节点。假设兴趣分组的转发路径为 College PK MD-Houston TX-Boulder CO-Salt Lake City UT。网络处于初始状态, 各个节点的缓存空间未被占用, 即 $I(v_i)$ 均等于 1。

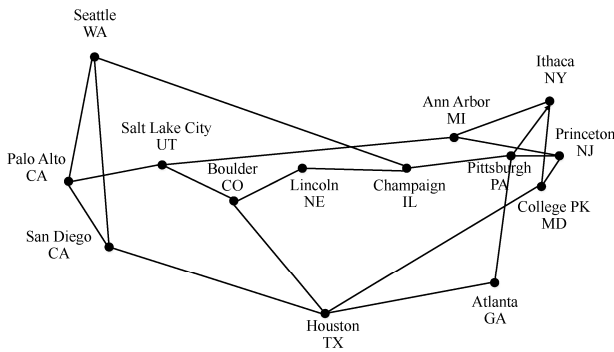


图 1 网络拓扑 NSFNET

在控制器端, 控制器会收集整个网络的拓扑信息和各个内容路由器的缓存空间空闲率, 计算各节点的度中心性 $C_D(v_i)$ 、紧密中心性 $C_C(v_i)$ 和介数中心性 $C_B(v_i)$ 并进行归一化处理, 确定 3 个度量所占的权重均为 $\frac{1}{3}$ (综合考虑各节点中心性度量, 不倾向于任一节点中心性度量), 并计算得出各个内容路由器的度量总分 $S(v_i)$, 再结合缓存空间空闲率 $I(v_i)$ 计算得到兴趣分组转发路径上各个内容路由器的缓存优先级 $P(v_i)$, 结果如表 1 所示。

表 1 转发路径上各内容路由器的计算结果

节点名称	$C_D(v_i)$	$C_C(v_i)$	$C_B(v_i)$	$P(v_i)$
MD	0.230 8	0.326 9	0.201 5	0.253 1
TX	0.307 7	0.307 7	0.421 2	0.345 5
CO	0.230 8	0.269 2	0.219 8	0.239 9
UT	0.230 8	0.269 2	0.245 4	0.248 5

从结果可以看出这 4 个节点缓存优先级的排序为 $P(TX) > P(MD) > P(UT) > P(CO)$, 若控制器根据业

务需求需要在路径上选取 2 个路由节点作为缓存节点, 则将选择节点 TX 和 MD, 并向其发送主动缓存内容的指令。当从提供者返回的数据分组途经节点 TX 和 MD 时, 会执行等待中的主动缓存指令, 将数据分组中内容缓存在节点 TX 和 MD 中。

3.3 CMC 的算法实现

下面, 是对 CMC 缓存机制的伪代码描述。算法首先根据控制器的全局视角获得全局网络拓扑 G' 和各内容路由器的缓存空间空闲率, 然后分别依据式(1)、式(2)和式(3)计算得到各个内容路由器的度 $C_D(v_i)$ 、 $C_C(v_i)$ 和 $C_B(v_i)$ 并进行归一化处理, 依据不同的需求确定 3 个度量所占的权重分别为 α 、 β 和 γ , 并依据式(5)计算得出各个内容路由器的度量总分 $S(v_i)$, 再依据度量总分 $S(v_i)$ 和缓存空间空闲率 $I(v_i)$ 通过式(6)计算得到兴趣分组转发路径上各个内容路由器的缓存优先级 $P(v_i)$, 最后对路径上各内容路由器的缓存优先级进行排序, 并按顺序依次选出 k 个内容路由器作为缓存节点并发出主动缓存指令。分析此算法可知其时间复杂度为 $O(n^3)$, 算法复杂度较大, 这主要是由于介数中心性计算带来的复杂度增加, 可以通过优化介数中心性的计算方式以及使用高性能的控制器降低其影响。

CMC 缓存算法如算法 1 所示。

算法 1 CMC 缓存算法

输入 $I = \{I_1, I_2, \dots, I_n\}$: 缓存空间空闲率

G' : 网络拓扑结构

$T = \{T_1, T_2, \dots, T_k\}$: 链路时延

输出 C : 缓存节点集

1) for ($i = 1; i \leq n; i++$)

2) $C_D(v_i) \leftarrow \frac{GetDegree(v_i)}{n-1}$; // v_i 的度中心性

3) $D(v_i) \leftarrow Dijkstra(v_i, G', T)$; // v_i 到其他所有节点的最短路径

4) $C_C(v_i) \leftarrow \frac{sum(power(2, -D(v_i)))}{n-1}$; // v_i 的紧密中心性

紧密中心性

5) for ($j = 1; j \leq n; j++$)

6) for ($l = 1; l \leq n; l++$)

7) $\sigma_{jl} \leftarrow Dijkstra(j, l, G', T)$; // 各节点对间所有最短路径

8) if $v_i \in \sigma_{jl}$

9) $C_B(v_i) \leftarrow C_B(v_i) + 1$; // 求包含节点 v_i 的路径条数

```

10)   end if
11)   end for
12)   end for
13)    $C_B(v_i) \leftarrow \frac{2C_B(v_i)}{n(n-1)}$ ; //  $v_i$  的介数中心性
14)    $S(v_i) \leftarrow \alpha C_D(v_i) + \beta C_C(v_i) + \gamma C_B(v_i)$ ; //  $v_i$  的
度量总分
15)   end for
16)    $P(v_i) \leftarrow \frac{I(v_i)S(v_i)}{\text{sum}(S)}$ ; //  $v_i$  的缓存优先级
17)    $P' \leftarrow \text{sort}(P)$ ; // 对缓存优先级进行排序
18)   for ( $j = 1; j \leq k; j++$ )
19)     for ( $i = 1; i \leq n; i++$ )
20)       if  $P(v_i) = P'(j)$ 
21)          $C \leftarrow v_i$ ; // 选出前  $k$  个节点作为缓存
节点
22)         break;
23)       end if
24)     end for
25)   end for
26)   return  $C$ 

```

4 CMC 性能评价

本节通过 ndnSim 建立仿真模型实现对处处缓存机制 LCE、随机缓存机制 Prob ($p=0.5$) 和基于节点中心性度量的缓存机制 CMC 的仿真,并在缓存命中率、内容获取跳数和平均请求时延 3 个评价指标上进行了定量的比较和分析。

4.1 评价指标

为了便于评价 CMC 缓存机制的性能,在此对相关评价指标做出定义。

1) 缓存命中率

缓存命中率 (CHR, cache hit ratio) 是指用户请求由缓存节点而不是源服务器响应的概率,它是反映缓存性能的典型参数,缓存命中率越高,缓存系统的效率越高,其计算式为请求被缓存节点命中的次数与用户发出的所有请求总数的比值。

2) 内容获取跳数

内容获取跳数 (CAH, content access hop) 是指用户请求到达内容服务节点(缓存节点或源服务器)的跳数,而缓存节点较源服务器离请求者近,所以具有更少的内容获取跳数。内容获取跳数越少,缓存系统的效率越高。

3) 平均请求时延

平均请求时延 (ARD, average request delay) 是指用户发出请求信息到返回内容所经历的平均时延。它可以反映请求的反应速度,而缓存节点较源服务器离请求者近,所以具有更快的反应速度。反应速度越快,说明缓存系统的缓存效率越高。

4.2 仿真设置

实验采用随机生成的包含 200 个节点和 600 条链路的网络作为时延拓扑,链路带宽设置为 10 Gbit/s。各节点的度中心性 $C_D(v_i)$ 、紧密中心性 $C_C(v_i)$ 和介数中心性 $C_B(v_i)$ 在总分中所占的权重均为 $\frac{1}{3}$ (综合考虑各节点中心性度量,不倾向于任一节点中心性度量)。用户的请求过程服从泊松分布,即请求的间隔时间服从指数分布,平均每单位时间 1~10 个兴趣分组。用户的请求模式服从 Zipf 分布,Zipf 参数越大,说明用户的偏好越集中,仿真时将 Zipf 参数设为 0.7。节点缓存替换策略选择 LRU。网络提供的内容数量为 1 000 个,用户数量为 100 个。假设每个节点缓存容量相同,均设为 m ,仿真时考虑相对缓存容量,即每个节点的缓存容量 m 与总的内容数量 1 000 的比值 ($\frac{m}{1000}$)。仿真过程中,将节点

缓存容量 m 由 1 增加到 100,相对缓存容量则由 0.01 增加到 0.1,缓存节点数 k 取 20。仿真的主要参数设置如表 2 所示。

参数	默认值	变动范围
节点数	200	—
链路数	600	—
链路带宽/(Gbit·s ⁻¹)	10	—
内容数量	1 000	—
用户数量	100	—
缓存节点数	20	—
用户请求模式	Zipf 分布: $\alpha=0.7$	—
用户请求过程	泊松分布: $\lambda=5$	1~10
节点缓存容量	均匀分布: $\mu=50$	1~100

4.3 仿真结果

为观察评价指标随某个参数的变化情况,以下各组实验均使观察的参数单个发生变化,其他参数保持不变,观察变化参数对评价指标的影响。

4.3.1 用户请求速率的影响

本节研究在不同的用户请求速率下 3 种缓存策略在缓存命中率、内容获取跳数和平均请求时延 3 个评价指标上的性能表现。用户请求过程服从泊松分布，速率变化范围为每单位时间 1~10 个。

从图 2 可以看出，随着用户请求速率的增加，3 种缓存机制的缓存命中率和内容获取跳数均无明显变化。但平均请求时延随用户请求速率的增加而逐渐增大。这是因为分组在 M/G/1 排队系统中所经历的时延与分组到达速率有关，当用户请求速率增大时，单位时间内注入网络的分组数量将增加，这使分组在排队系统中的排队时延增加，进而增加平均请求时延。对比分析可以看出，CMC 在 3 项评价指标上均优于另外 2 项缓存机制。这表明，在相同的用户请求速率下，相比于处处缓存机制 LCE 和随机缓存机制 Prob ($p=0.5$), CMC 具有较高的缓存性能。

4.3.2 节点缓存容量的影响

本节研究在不同的节点缓存容量下 3 种缓存策略在缓存命中率、内容获取跳数和平均请求时延 3 个评价指标上的性能表现。节点缓存容量服从均匀分布，变化范围为 1~100 个，则相对缓存容量的变化范围为 0.01~0.1。

从图 3 可以看出，随着相对缓存容量的增加，3 种缓存机制的缓存命中率逐渐增大，而内容获取跳数和平均请求时延均逐渐减小。对比分析可以看出，CMC 在 3 项评价指标上均优于另外 2 项缓存机制。这表明，在相同的相对缓存容量下，相比于处处缓存机制 LCE 和随机缓存机制 Prob (0.5)，CMC 具有较高的缓存性能。

4.3.3 仿真结果分析

通过实验可以看出，当用户请求速率和节点缓存容量变化时，CMC 在缓存命中率、内容获取跳数和平均请求时延 3 个评价指标上的性能表现均优

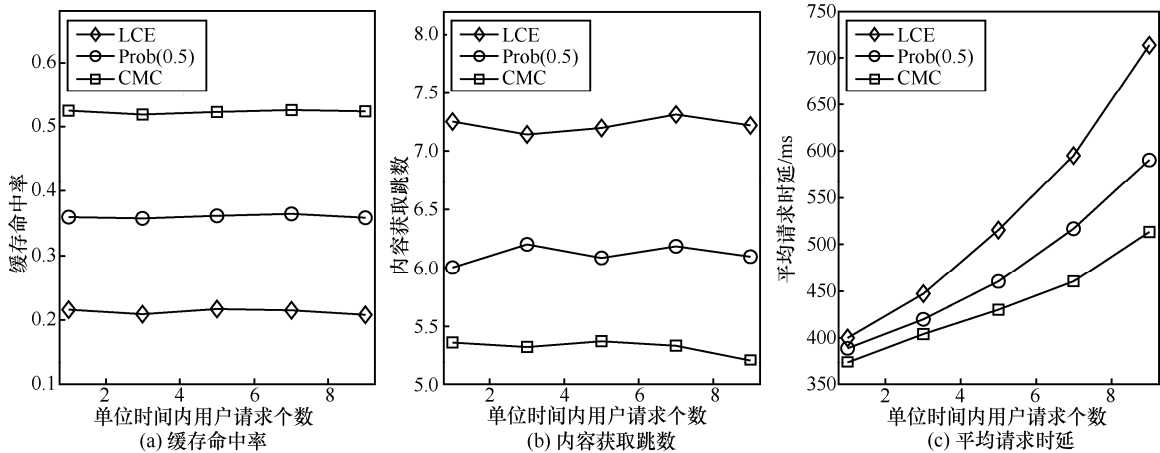


图 2 用户请求速率对评价指标的影响

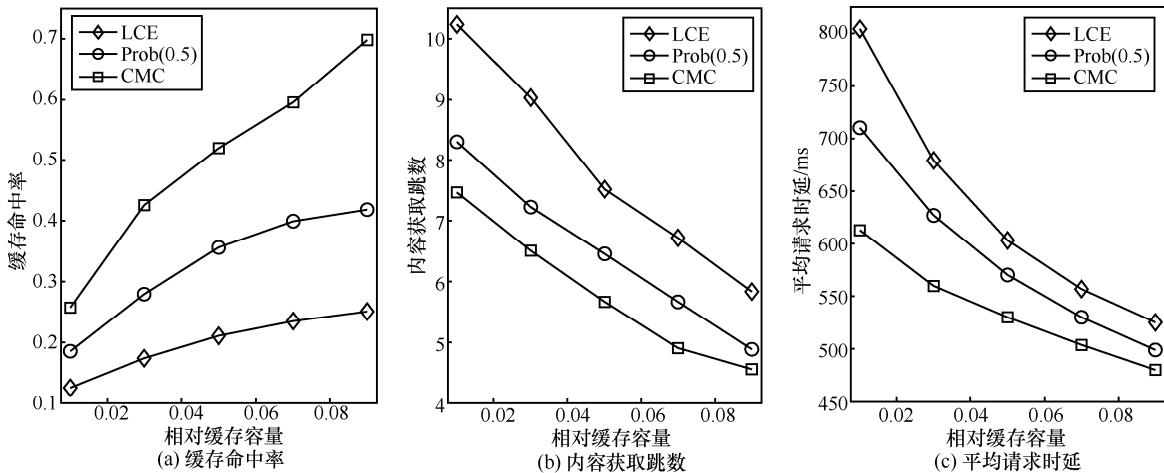


图 3 节点缓存容量对评价指标的影响

于处处缓存策略 LCE 和随机缓存策略 Prob(0.5)。这是因为 LCE 要求路径上的所有节点无差别地对数据分组中的内容资源进行缓存, 产生大量缓存冗余, 路由节点上缓存的内容多样性较低, 缓存性能极差; Prob 要求数据分组返回路径上的所有路由节点都以固定的概率缓存对象, 虽然能减小缓存空间占用率, 但产生大量缓存冗余, 缓存的内容多样性不足, 缓存性能较差; CMC 可以通过控制器利用全局视角选择转发路径上具有较高的度中心性、紧密中心性和介数中心性的节点作为缓存节点, 这些节点有较大的概率出现在兴趣分组的转发路径上, 且 CMC 机制考虑了节点的缓存空间空闲率, 可以有效地利用节点的缓存空间, 避免部分关键节点上缓存内容的频繁替换。因此, 相对于其他缓存机制, CMC 具有较高的缓存性能。

4.4 性能分析讨论

从节点的介数中心性的计算过程中可以发现, 其计算复杂性较大, 但各节点的中心度量的计算只需要在机制运行的初期完成即可。后续针对不同的内容, 只需要获取各节点的缓存空间空闲率即可完成对缓存节点的选取。因为只要网络拓扑稳定, 无频繁的节点删除或添加, 网络中各节点的中心性度量就不会发生改变, 可被重复利用。节点中心性度量计算所带来的时延影响只会发生在机制运行初期及网络拓扑发生变化的时候。

由于 CMC 机制所选出的缓存节点很大概率为网络的核心节点, 这将为这些节点带来较大负载, 后续考虑将内容放在核心节点邻居的放置方式, 以减少核心节点的负担。同时, 本文在对各节点中心性度量进行融合时, 考虑了选用单个节点中心性度量的效果是否比 3 个节点中心性度量进行融合要好, 所以为 3 个节点中心性度量均设定了权重因子, 对 3 个权重因子中的 2 个因子进行置零操作就可达到使用单个节点中心性度量的效果, 但在实验过程中发现, 随着网络拓扑结构的变化, 不同的节点中心性度量的使用对不同网络拓扑结构的影响不同, 这是由于每个节点中心性度量所考量的方向有所不同, 度中心性考量的是其邻居的个数, 紧密中心性考量的是节点传播信息时不依靠其他节点的程度, 介数中心性考量的是节点控制信息传输的能力, 所以本文考虑融合 3 种节点中心性度量来对节点的重要性综合考量, 以此实现对不

同拓扑的普适。

5 结束语

本文主要研究了 CCN 的缓存机制, 提出了一种基于节点中心性度量的内容中心网络缓存机制 CMC。CMC 利用控制器的全局视角获得全局的网络拓扑和各个内容路由器的缓存空间空闲率, 计算得到各个内容路由器的紧密中心性和介数中心性并进行归一化处理, 再利用加权得出各个内容路由器的度量总分, 并综合考虑缓存空间空闲率得出缓存优先级, 最后对路径上各内容路由器的缓存优先级进行排序, 并按顺序依次选出若干内容路由器作为缓存节点并发出主动缓存指令。仿真实验表明: 与处处缓存机制 LCE 和随机缓存机制 Prob(0.5) 相比, CMC 可以有效地提高缓存命中率, 降低内容获取跳数和平均请求时延。

由于本文选出的节点很大概率为网络的核心节点, 为减小网络核心节点的负担, 后续会继续探究将内容放置在核心节点邻居的可能性。同时, 也将就本文机制对网络拓扑的敏感性问题进行研究, 希望在整个机制运行之前, 对网络拓扑的特征进行提取, 据此在机制运行过程中调整 3 个权重因子的大小, 以达到更优的性能。

参考文献:

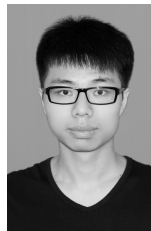
- [1] KOPONEN T, CHAWLA M, CHUN B, et al. A data-oriented (and beyond) network architecture[C]//ACM SIGCOMM. 2007: 181-192.
- [2] FOTIOU N, TROSSEN D, POLYZOS G. Illustrating a publish-subscribe internet architecture[J]. Telecommunication Systems, 2012, 55(4): 233-245.
- [3] DANNEWITZ C, KUTSCHER D, OHLMAN B, et al. Network of information (NetInf)-an information-centric networking architecture[J]. Computer Communications, 2013, 36(7): 721-735.
- [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] 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.
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.
- [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. 2012: 55-60.
- [8] LI Z, SIMON G. Time-shifted TV in content centric networks: the case for cooperative in-network caching[C]//In IEEE International Conference on Communications. 2011: 1-6.
- [9] MING Z X, XU M W, WANG D. Age-based cooperative caching in information-centric networks[C]//IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPs). 2012: 268-273.
- [10] HE D L, CHAI W K, PAVLOU G. Leveraging in-network caching for efficient content delivery in content-centric network[C]//In London Communication Symposium, 2011.
- [11] CHAI W K, HE D L, PSARAS I, et al. Cache “less for more” in information-centric networks[J]. Computer Communications, 2013, 36(7): 758-770.
- [12] CHANDA A, WESTPHAL C. A content management layer for software-defined information centric networks[C]//In ACM SIGCOMM Workshop on Information-Centric Networking, 2013: 47-48.
- [13] WANG J, REN J, LU K J, et al. An optimal cache management framework for information-centric networks with network coding[M]//IFIP Networking Conference. 2014: 1-9.
- [14] MENDONCA M, NUNES B A, NGUYEN X, et al. A survey of software-defined networking: past, present, and future of programmable networks[J]. IEEE Surveys Communications and Tutorials, 2014, 16(3):1617-1634.
- [15] DU Y, GAO C, HU Y, et al. A new method of identifying influential nodes in complex networks based on TOPSIS[J]. Physica A Statistical Mechanics & Its Applications, 2014, 399(4):57-69.
- [16] PODLIPNIG S, BÖSZÖRMENYI L. A survey of web cache replacement strategies[J]. ACM Computing Surveys, 2003, 35(4):374-398.
- [17] JUAN D, ROGERS. Internetworking and the politics of science: NSFNET in Internet history[J]. The Information Society, 1998, 14(3): 213-228.

作者简介:



蔡岳平 (1980-), 男, 江苏丹阳人, 重庆大学副教授、硕士生导师, 主要研究方向为数据中心网络、光通信网络、未来互联网等。



刘军 (1990-), 男, 江西遂川人, 重庆大学硕士生, 主要研究方向为内容中心网络、软件定义网络等。



樊欣唯 (1992-), 女, 重庆人, 重庆大学硕士生, 主要研究方向为内容中心网络、网络功能虚拟化。