

# 未来网络中 CCN 的缓存策略的研究

杨超, 杨龙祥

(南京邮电大学通信与信息工程学院, 江苏南京 210000)

**摘要:**为了克服现有 Internet 存在的众所周知的缺点,未来网络成为研究热点,而以内容为中心的网络(CCN)在众多方案中被认为是最有前途的方案之一。CCN 网络采用了分布式的内容缓存机制,允许节点对传输的内容进行缓存。而缓存放置策略是研究的热点。文中在对几种常见的缓存决策方案进行分析的基础上,分析存在的问题,重点研究了一种动态缓存决策方案,根据内容的流行度规律和用户的潜在需求,在必要的节点上选择性地缓存。然后在 ccnSim 上进行仿真对比,实验结果表明,缓存效率明显高于基本的缓存策略。

**关键词:**未来网络;以内容为中心的网络;缓存放置;流行度

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2015)05-0064-04

doi:10.3969/j.issn.1673-629X.2015.05.016

## Research on Cache Placement Strategy for CCN in Future Network

YANG Chao, YANG Long-xiang

(Department of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210000, China)

**Abstract:** In order to overcome the well known deficiencies of the current Internet, the future network is becoming the research hotspots, and great expectations are held on the Content-Centric Network (CCN). A distributed cache strategy which allows the network node to store the content is used in CCN. The cache placement of CCN is a popular issue now. With the analysis of some common cache placement strategies, propose a dynamic cache placement strategy based on the popularity of the content and potential demands. Content will be cached on some nodes. Then, simulation experiments are conducted on ccnSim. The experimental results show the cache efficiency will be obviously improved.

**Key words:** future network; CCN; cache placement; popularity

### 0 引言

随着传统 TCP/IP 网络诸多弊端的暴露,以内容为中心的网络(Content-Centric Network, CCN)<sup>[1]</sup>作为一种新型网络架构,已经成为未来网络<sup>[2]</sup>的研究热点。CCN 将内容置于首要地位,不再关心内容的存储位置,而仅关心内容本身,内容的来源可以是网络中的内容服务器,也可以是任意网络节点中的缓存。当用户需要获取内容时,首先作为内容请求者,向网络发送内容请求,然后根据 CCN 的转发策略,对内容请求进行逐跳转发,任何能够响应内容请求的节点或内容服务器都可以沿着内容请求的方向,直接将内容传送给请求者,完成信息的传输。CCN 网络采用了分布式的内容缓存机制,允许节点对传输的内容进行缓存。尽管

缓存技术已经广泛应用于计算机网络领域,但考虑到 CCN 中缓存功能的普遍性和通用性,缓存技术的研究仍存在很多需要解决的问题。目前 CCN 缓存技术的研究主要涉及缓存容量的分配策略、缓存替换策略、缓存利用策略和缓存放置策略<sup>[3]</sup>,其中缓存放置策略是 CCN 缓存研究的核心问题。但是当前提出的缓存机制还是有一些问题需要解决:

首先是缓存无效,当很多节点上缓存相同的内容时,有一些节点,比如上游节点上的内容在一段时间内从未被请求过,而导致这些缓存一直未发挥作用,这就叫无效缓存。这会导致空间的浪费。

其次是相同的缓存内容在某一段时间内存在于各个节点,尤其是在热门时间内,各个节点上都缓存着同

收稿日期:2014-06-29

修回日期:2014-09-29

网络出版时间:2015-04-22

基金项目:国家“973”重点基础研究发展计划项目(2013CB329104);国家自然科学基金资助项目(61372124)

作者简介:杨超(1990-),男,硕士研究生,研究方向为未来网络智慧缓存技术、无线传感器网络;杨龙祥,教授,博士,博士后,博士生导师,研究方向为无线通信系统和物联网。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20150422.1005.013.html>

一内容,而当热门时间一过,这些内容又几乎同时消失了。

以上两点导致缓存效率不高。

为了解决上述的缓存冗余和缓存同质,文中在分析几种基本缓存策略的基础上,提出了基于内容流行度的缓存决定策略,并分析研究了其效率和优点。

### 1 CCN 的结构原理

#### 1.1 CCN 的节点结构

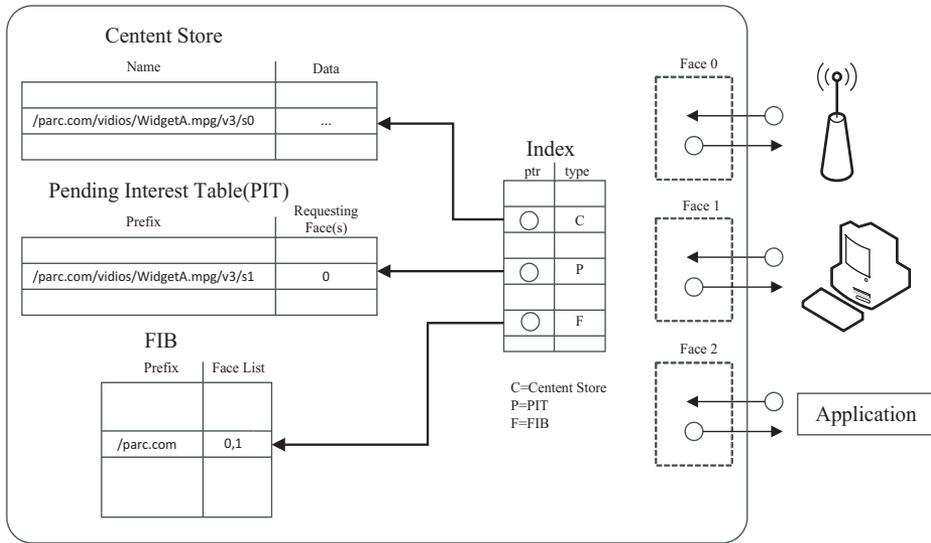


图1 CCN 网络节点图

#### 1.2 CCN 的包处理过程

当一个兴趣包到达某节点之后,首先查找 CS 中是否存在需要的数据包,如果存在,则将数据包待会请求节点,路径是兴趣包到达该节点的路径。如果 CS 中不存在请求的内容,则去 PIT 中查找,如果兴趣包的内容前缀与 PIT 的内容前缀匹配,则将兴趣包丢弃,并且 PIT 响应条目中增加内容到达的接口<sup>[5]</sup>。如果 PIT 中没有与 CS 匹配,则去查找 FIB,将兴趣包利用洪范的形式向除了兴趣包到达的接口以外的其他接口转发,并且将兴趣包最终到达的接口记录在 PIT 里。如果在 FIB 中也没查找到,则丢弃兴趣包。

当数据包到达某个节点时,需要对数据包的内容命名字段进行最长前缀匹配。首先查询 CS,如果已经存在该数据包,则丢弃;如果没有,再与 PIT 条目匹配,如果 PIT 中有匹配的条目,则向相应端口转发数据包,并且缓存至 CS,如果 PIT 中没有匹配的条目,则丢弃数据包。

### 2 基本的 CCN 缓存决定策略

缓存决定策略是用来决定哪些内容或者数据包应该被缓存,而哪些内容不需要被缓存。缓存决定策略已经成为未来网络的一个研究热点。目前在 CCN 中

CCN 的节点结构<sup>[4]</sup>如图 1 所示,主要包括 CS (Content Store),PIT(Pending Interest Table),FIB(Forward Information Base)三个部分,即内容存储器、未决请求表和转发表三个部分。而在 CCN 的通信过程中,存在两种包类型:兴趣包和数据包。兴趣包携带内容请求,数据包携带被请求的内容。所以对于 CCN 的节点,CS 用来缓存节点收到的内容,PIT 记录未响应的兴趣包以及响应到达的端口,FIB 则类似于 IP 路由器的作用,用来保存兴趣包的下一跳接口。

应用最多的缓存决定策略是 LCE(Leave Cache Everywhere),即处处缓存策略,以及由此引申的 LCD(Leave Copy Down)和 MCD(Move Copy Down)。

#### 2.1 处处缓存策略(Leave Cache Everywhere)

内容会被缓存在返回路径的所有节点上<sup>[6]</sup>。如图 2 所示,当节点 H 从服务器请求了内容 o 之后,中间节点 A 和 B 都会缓存内容 o,所以当节点 E 或者 F 再次请求内容 o 时,会直接从 A 或者 D 获取,此时节点 B 的缓存失去了作用,所以导致了缓存的冗余。

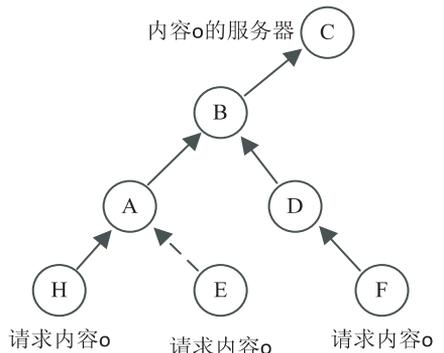


图2 LCE 示意图

#### 2.2 向下拷贝策略(Leave Copy Down)

如图 3 所示,当节点 A 的兴趣包命中即找到需要的数据包以后,该方案只允许把数据包缓存在数据包

返回途中下游节点 B 上。这也就避免 LCE 那样的相同数据包的大量拷贝。当响应了多个请求后,中间节点的内容就会被逐渐推向网络边缘。所以当再次有请求时,总是离用户更近的边缘节点首先响应,减少了用户网络响应用户请求的时间。但是,流行度高的内容,需要被访问多次才会被缓存到边缘节点,同样会产生大量的内容冗余备份<sup>[7]</sup>。

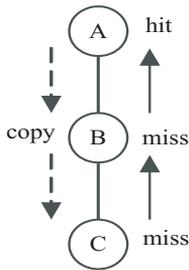


图 3 向下拷贝策略

### 2.3 向下拷贝删除策略 (Move Copy Down)

如图 4 所示,当节点 A 的兴趣包命中以后,将内容缓存到节点 B,此时节点 A 上缓存的数据包也会被删除,所以只有节点 B 上存在缓存。因此 MCD 的冗余度比 LCD 大大降低。因为当 A 节点满足兴趣包并且把兴趣包缓存到下游节点后,数据包已经在向网络边缘靠拢,再次有对此数据包的请求时,节点 B 会先响应,节点 A 的利用率已经不高了,可以被删除。但是 MCD 中内容缓存点是动态变化的,内容缓存节点动态地往边缘扩散,这样会导致额外的网络开销<sup>[8]</sup>。

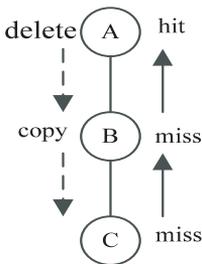


图 4 向下拷贝删除策略

### 3 基于内容流行度的动态缓存策略

基于上述几个基础方案存在的问题,文中提出一种基于内容流行度的动态缓存决定策略。基本原理如下:传统的 LCE 或者 LCD 等对任何内容都会进行缓存,而没有考虑到内容的流行度,因为有的内容较为流行,需求量较大,而有的内容需求量较小。所以应该尽量多缓存流行度较高的内容,而流行度较低的内容,则尽量少缓存。这样可以节省资源,并且提高了 CCN 的整体表现能力。将这个策略叫做 MPC (Most Popular Content),即只缓存流行内容,而不缓存不流行的内容,不缓存不流行的内容可以减少缓存替换时的负担,因为缓存的不流行的内容越多,那么当流行的内容被

请求时,不流行的内容就会被替换下来(用 LRU 等策略),这样会增加缓存替换的工作量。

那么流行度怎么定义呢<sup>[9]</sup>? 笔者定义了一个流行阈,当一个内容被请求了一定的次数,而达到了这个流行阈之后,这个内容就被称为流行的内容。但是一个内容可能在一定时间内请求的节点比较多,但是过了一段时间后,渐渐变少了。所以将流行度定义为一定时间内被请求的次数,这个时间定义为统计周期或者是重置时间。比如,如果统计周期为 5 s,当内容 o 在 5 s 内被订阅了 3 次之后,它的流行度就是 3。当时间过了 5 s 之后,它的流行度就需要被重新计数了。所以当内容 o 刚刚被发布时,订阅它的节点很多,所以在几个统计周期内它的流行度都很高。但是在过了一段时间之后,订阅者逐渐减少。而且每隔一段时间流行度都需要被清零,所以流行度也就逐渐下降。在每个节点上,都会维护着一个流行度表<sup>[10]</sup>,并且记录着它收到的请求特定内容的兴趣包。当一个内容或者说数据包变得“流行”以后,缓存该内容的节点,会通知它的相邻节点,让它的相邻节点也缓存这个内容。但是收到建议后,相邻节点可以做出选择,可以缓存这个内容,也可以不缓存。需要注意的是,内容的流行度不是一直不变的,当一个内容变成流行内容之后,它可以在一段时间后变得不流行了,因为用户的需求随时在变化。

下面根据具体的实例来分析基于内容流行度的缓存决策方案。如图 5 所示,在一个统计周期内,最开始在节点 D 缓存着一个流行内容  $d_1$  而节点 E 缓存着一个不流行的内容  $e_1$ ,假设为 0。当节点 A 发出请求内容  $e_1$  的兴趣包之后,包括节点 A, C, D, E 在内的沿路节点中,  $e_1$  在流行度表中的流行度都上升了。也就是说,  $e_1$  在节点 A, C, D, E 中的流行度现在已经由 0 变为 1 了。同时,当节点 A 发出请求内容  $d_1$  的兴趣包之后,包括沿路节点 A, C, D 在内的节点,  $d_1$  在流行度表中的流行度也变为 1 了。接下来,节点 B 请求内容  $d_1$ ,则在节点 C 和 D 中,  $d_1$  的流行度变成了 2。最后 C 请求内容  $d_1$ ,那么在节点 C 和 D 中,内容  $d_1$  的流行度也就变成了 3。在这里把阈值设置为 3,即流行度大于等于 3 的内容会被缓存。此时,节点 D 是唯一缓存内容  $d_1$  的节点, D 向它的相邻节点 E 和 C 发送消息,告诉它们,可以缓存内容  $d_1$ 。至于是否缓存,还要看节点 C 和 E 的缓存能力。因为每个节点中,另外维护了两个参数,最大缓存数和应当缓存数目,当缓存能力足够时,就会缓存内容  $d_1$ 。所以当 E 和 C 缓存了  $d_1$  之后,比如当节点 A 请求内容  $d_1$  时,就会直接从节点 C 处获取,而不是节点 D<sup>[11]</sup>。

若此时用 LCE 的传统方法,当节点 A 请求  $e_1$  时,

节点 A,C,D,E 都将缓存  $e_1$ , 而节点 A 请求  $d_1$  时, A,C, D 均会缓存  $d_1$ 。而基于 MPC 的方案最终只有节点 C 和 D 缓存了  $d_1$  而节点 E 缓存了  $e_1$ 。所以相比于传统的缓存方案, 节省了很多的缓存空间, 也降低了很多缓存替换需要做的工作<sup>[12]</sup>。

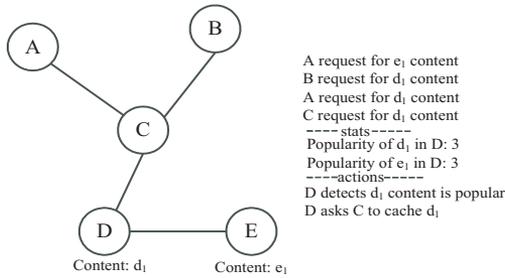


图5 MPC 示意图

### 4 仿真

为了分析 MPC 的性能, 利用 ccnSim 仿真器进行了仿真。ccnSim 是一个 C++ 基于 Omet++ 框架开发的仿真器。在 ccnSim 中认为用户对内容的偏好是服从 Zipf 分布的。在每次仿真中, 都会设置 1 个数据字典 (原始内容服务器) 和 8 个数据请求节点。数据字典的大小被设置为 1 PB, 而每个节点的缓存大小被设置为 10 GB。每次这 1 个数据字典和 8 个数据请求节点会被随机放入 ccnSim 内置的拓扑图之一, 比如 Tree、Tiger2、Geant 等 6 个。每次仿真都会进行 10 次, 然后取平均值。

这里仿真对比的是 LCE 和 MPC 两个缓存策略, 而缓存置换策略用的都是 LRU, LRU 就是把在缓存中最近最少使用的缓存给置换掉。MPC 和 LCE 的性能比较可用很多参数衡量, 这里用的是 Cache Hit Ratio, 缓存命中率, 意思就是沿路获得缓存命中的概率。显然, Cache Hit Ratio 越大, 说明缓存效率越高。将所有 10 次的平均值进行绘制, 如图 6 所示。

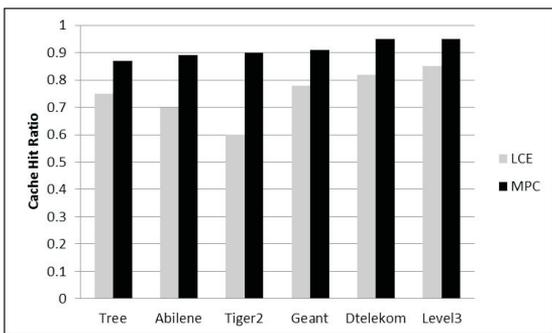


图6 MPC 与 LCE 仿真对比

可以看出, MPC 的缓存命中率一直在 85% 以上, 而 LCE 的缓存命中率一直低于 MPC。在 Level3 这个拓扑结构中, LCE 的缓存命中率达到最高, 但是还是低于 MPC 的缓存命中率。所以从图中可见。MPC 缓存

策略的缓存效率一直高于 LCE, 并且在某些拓扑结构下还高了很多。所以 MPC 缓存决策优于简单的缓存决策。

### 5 结束语

为了分析研究 CCN 网络的缓存决策策略, 文中分析研究了几种基本的缓存决策策略, 如 LCE、LCD 等, 得出了它们存在的缺点和不足, 比如无效缓存、冗余度高等等。在此基础上提出了一种基于内容流行度的缓存决策策略, 将内容流行度引入了缓存决策分析。经过仿真分析, 基于内容流行度的缓存决策在缓存效率上远远优于简单的 LCE 缓存决策。展望未来, 还有很多地方值得研究, 比如将 MPC 推动到移动网络环境中去, 还可以利用节点之间的带宽来换取缓存效率的思想<sup>[13]</sup>。

#### 参考文献:

- [1] 李军, 陈震, 石希. ICN 体系结构与技术研究[J]. 信息安全学报, 2012(4): 75-80.
- [2] 吴建平, 林嵩, 徐格, 等. 可演进的新一代互联网体系结构研究进展[J]. 计算机学报, 2012, 35(6): 1094-1108.
- [3] 杜永斐. 内容分发系统中关键策略研究[D]. 成都: 电子科技大学, 2013.
- [4] Wang J. A survey of web caching schemes for the Internet [J]. ACM SIGCOMM Computer Communication Review, 1999, 29(5): 36-46.
- [5] 唐丽均, 李云, 柴毅, 等. 一种结合传染路由的缓存调度算法[J]. 西安电子科技大学学报, 2012, 39(1): 141-145.
- [6] 李春洪, 冯国富, 顾铁成, 等. 一种无热点的覆盖网协同缓存策略[J]. 软件学报, 2008, 19(3): 744-754.
- [7] 李文中, 陈道蓄, 陆桑璐. 分布式缓存系统中一种优化缓存部署的图算法[J]. 软件学报, 2010, 21(7): 1524-1535.
- [8] 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.
- [9] Rossi D, Rossini G. Caching performance of content centric networks under multi-path routing (and more) [R]. [s. l.]: [s. n.], 2011.
- [10] Perino D, Varvello M. A reality check for content centric networking [C] // Proc of ACM SIGCOMM ICN Workshop. [s. l.]: [s. n.], 2011: 44-49.
- [11] Muscarello L, Caroglio G, Gallo M. Bandwidth and storage sharing performance in information centric networking [C] // Proc of ACM SIGCOMM ICN workshop. [s. l.]: [s. n.], 2011: 26-31.
- [12] 王楠. 基于 CCN 的路由优化策略研究[D]. 北京: 北京交通大学, 2014.
- [13] 林闯, 贾子晓, 孟坤. 自适应的未来网络体系架构[J]. 计算机学报, 2012, 35(6): 1077-1093.

# 未来网络中CCN的缓存策略的研究

作者: [杨超](#), [杨龙祥](#), [YANG Chao](#), [YANG Long-xiang](#)  
作者单位: [南京邮电大学 通信与信息工程学院, 江苏 南京, 210000](#)  
刊名: [计算机技术与发展](#)   
英文刊名: [Computer Technology and Development](#)  
年, 卷(期): 2015(5)

引用本文格式: [杨超](#). [杨龙祥](#). [YANG Chao](#). [YANG Long-xiang](#) [未来网络中CCN的缓存策略的研究](#)[期刊论文]-[计算机技术与发展](#) 2015(5)