

内容中心网络及其缓存策略研究

段 炼,杨龙祥,任美翠

(南京邮电大学 通信与信息工程学院,江苏 南京 210003)

摘 要:当今,因特网的使用已经演进到以内容的分发和检索为主,但是目前使用的 IP 结构并不能很好地满足这样的需求。内容中心网络(Content Centric Networking,CCN)打破了传统的“主机—主机”的通信模式,将内容置于核心地位,是目前未来网络领域的研究热点之一。CCN 将内容与主机分割开并将内容存储在网络节点中。任何存有内容的节点都可以充当服务器向用户提供服务,因此 CCN 可以高效地进行内容传输。CCN 的这种优势来自于它可以进行网络内缓存,可以说缓存是 CCN 的基石。缓存策略的性能直接影响了整个 CCN 的性能。简要概括了 CCN 和 IP 的区别,介绍了 CCN 的一些核心概念和工作机制,给出了 CCN 中相关的缓存算法并分析了它们的优缺点。

关键词:内容中心网络;未来网络;网络体系结构;缓存算法

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2017)03-0075-06

doi:10.3969/j.issn.1673-629X.2017.03.016

Research on Content Centric Networking and Its Caching Strategies

DUAN Lian,YANG Long-xiang,REN Mei-cui

(College of Communication and Information Engineering,Nanjing University of
Posts and Telecommunications,Nanjing 210003,China)

Abstract:Use of Internet in today's world has evolved to be dominated by distribution and retrieval of content,while currently used IP architecture cannot meet this demand. Content Centric Networking (CCN) breaks the traditional "host to host" communication mode,making content in the core position is one of the research hotspots. CCN decouples content from host and stores the content in every node. Any node can act as host and serve client if the requested content has been cached in it. Thus,CCN can deliver content efficiently. These advantages are mainly based on that CCN supports in-network caching,so it can be concluded that caching is backbone of CCN. The performance of the caching strategies directly affects the performance of CCN. The difference between the CCN and IP is summarized in brief,and then the key concepts and working mechanism of CCN are introduced. Finally,some proposed caching strategies have also been covered along with analyzing their advantages and disadvantages.

Key words:content centric networking;future network;network architecture;caching algorithm

1 概 述

在计算机系统开发的初期,系统内的资源非常有限。随后,为了解决资源共享的问题,不同的网络结构逐渐演化,最终形成了客户端—服务器的通信模式^[1]。在这种模式中,一个节点拥有资源并且其他节点可以利用这些资源。这是一种以主机为中心的面向连接的网络模型。在这种模式中,为了从一个特定服务器获取相应服务,节点必须与此服务器进行连接。计算机系统发展至今,系统内的资源不再匮乏且变得相对廉价,加上技术的进步,使得这种模式在逐渐转变。

计算机体系结构设计的初衷是用于共享资源,但

是随着网络的不断发展,资源的共享已经不能满足人们的需求。现今,人们对于内容的传播更为感兴趣(主要是视频),而这也占据了因特网中大部分的流量。TCP/IP 正是为了应对内容的传播而设计的。

例如,考虑到图 1 中的网络拓扑,其网络结构使用的是 TCP/IP。因此网络中的每一个节点将遵循 TCP/IP 协议。在此情况下,如果节点 1 需要获取内容 A,那么此节点首先需要和提供内容 A 的服务器进行连接。因此,节点 1 需要向服务器发送一个连接请求。如果此时服务器可以提供服务,那么它将会接收连接请求。之后节点 1 和服务器之间将会建立连接。现在,节点

收稿日期:2016-04-22

修回日期:2016-08-10

网络出版时间:2017-02-17

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

作者简介:段 炼(1991-),男,硕士,研究方向为移动通信与无线技术;杨龙祥,教授,博士生导师,研究方向为移动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.20170217.1628.036.html>

1 发送需求内容 A 的请求,而服务器将会把内容 A 转发给节点 1 来完成响应。

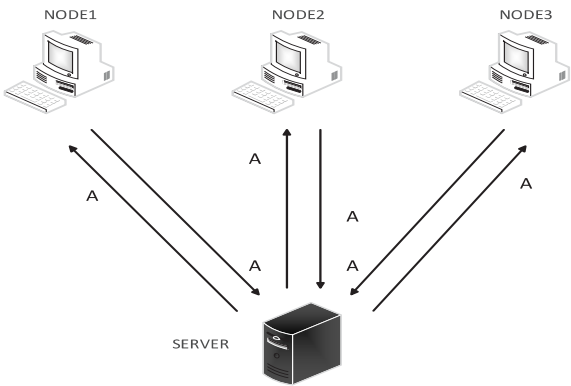


图 1 IP 结构的低效性

同样,如果节点 2 和节点 3 也需要内容 A,那么它们将会执行和上面一样的操作。可以说,无论多少节点需要内容 A,它们都需要进行相同的操作,即和提供内容的服务器进行连接。可以看出,IP 网络以主机为中心,它更加关注被请求信息的位置而不是信息本身的内容,因此,所有对于相同内容的请求都需要源服务器来完成,不仅导致网络的效率不高,还造成了巨大的带宽浪费。内容中心网络(CCN)^[2]正是为了克服 IP 网络的这种缺陷而提出的新型网络架构。

CCN 支持网络内缓存,即在网络内的所有节点都设有缓存功能^[3]。通过网络内缓存,CCN 避免了对相同内容的重复传输。当转发的内容经过某一节点时,此节点可以缓存该内容。当此节点再次收到对这一内容的请求时,它可以直接将请求的内容发送给请求者而不再需要请求服务器。因而 CCN 节省了网络资源,加快了内容传输。

文中主要对内容中心网络及其缓存策略进行研究,分析了 CCN 体系结构与 IP 体系结构的不同,介绍了 CCN 的包类型、节点转发的工作机制,阐述了几种典型的 CCN 缓存决策策略和缓存替换策略。在此基础上,重点研究了几种具有代表性的缓存决策策略,并对其性能的优缺点进行了分析,为未来的研究工作奠定了理论基础。

2 内容中心网络中的核心概念

与传统的主机到主机的网络结构不同,CCN 是一种面向内容的结构。当今用户的需求已经由传统的“资源共享”向“内容传播”转变,这正是 CCN 被提出的内在动力。CCN 是一种全新结构,它与传统的 IP 结构截然不同。图 2 在协议栈方面对 CCN 和 IP 进行了比较。

可以看出,IP 中的大部分层都需要双边协议,例如 IP 的 2 层框架协议是两个物理终端之间的协议,4

层传输协议是生产者和消费者之间的协议,而只有第 3 层即网络层需求统一的协议。CCN 的第 3 层即策略层有效地利用了多径连接。CCN 采用接收端驱动的方式,保证了传输内容的安全性,避免了许多一直困扰 IP 网络的安全隐患^[4]。

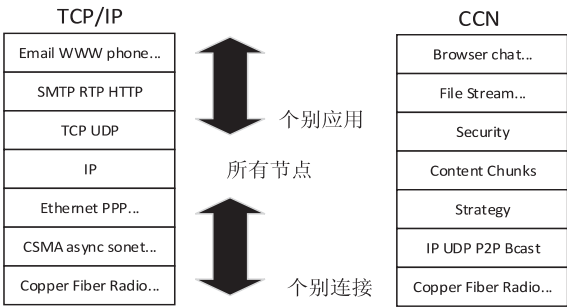


图 2 IP 和 CCN 协议栈

在 CCN 中,传入流量一般遵循 Zipf 分布^[5]。这种分布为流行的内容分配相应的等级。流行度表示某一内容在一定时间内被访问的次数。被访问的次数越多意味着这个内容的流行度越高。Zipf 分布依据流行度对内容进行分类并给内容分配相应的等级。内容的流行度越高其等级越低,流行度越低等级也就越高。

在 CCN 中任何的通信都由用户自己管理,而不再由传统的服务器进行管理。用户只会接收到那些已经需求过的内容。然而,在 IP 网络中,除了那些被需求的内容外,一些未被需求的内容也可能随之一起发送给用户。显然,相比 IP 网络,CCN 更具有安全性。

CCN 结构建立在命名数据的基础上,使用分级命名来唯一地命名内容。在 CCN 中有两种包结构:Interest 包和 Data 包。在 CCN 中,通过 Interest 包中的内容名称进行内容的请求。如果用户需求某个内容,他将会广播 Interest 包,任何节点在收到 Interest 包后,如果在此节点内存有用户请求的内容,那么此节点将向用户返回相应的 Data 包。当 Interest 包的内容名与 Data 包内容名的前缀相符合时,则该 Data 包满足该 Interest 包的请求。在 CCN 中,一个 Interest 包只会收到一个 Data 包作为响应,这确保了流量的平衡以及在节点之间高效的通信。

由于网络的高度动态性,Data 包和 Interest 包都有可能出现丢失的情况。因此,为了确保 CCN 的可靠性,当需求在一段时间内未被满足时,请求节点需要重发 Interest 包。

3 CCN 工作机制

CCN 致力于对数据的高速获取。为了达成这个目标,CCN 采用了新的转发引擎模型。其中包含三个主要的数据结构:转发信息表(Forwarding Information Base, FIB)、内容存储表(Content Store, CS)和待定请求

表(Pending Interest Table, PIT)。

FIB 将 Interest 包转发到包含有请求内容的服务器。CCN 的 FIB 和 IP 的 FIB 类似,不同的是 CCN 的 FIB 有一系列的出口,而 IP 的 FIB 只有一个^[6]。因此 CCN 中的节点可以向多个源服务器发送请求。CCN 中的 CS 和 IP 中的缓冲区域类似,用来作为缓存内容的存储区域。但 CS 有着不同的缓存替换策略。IP 缓冲区中的内容在转发之后将被丢弃,而 CCN 中的 CS 将会尽可能长时间地缓存内容,以便更快地为之后的请求提供服务。PIT 记录已经转发的 Interest 包,当对同一内容有多个请求时,它将多余的请求记录在一个条目中,并只向数据源发送一条请求。当 Data 包返回时,它会根据 PIT 中记录的条目将数据发送给相应的请求源。

3.1 Interest 包的处理

当一个 Interest 包到达一个节点的接口之后,节点将会对其进行最长前缀匹配来验证它所需的内容是否已经存储在 CS 中。如果需求的内容已经存储在节点的 CS 中,那么此节点将立刻向请求者发送 Data 包。由于请求已经被满足,所以此 Interest 包将会被丢弃。

如果在 CS 中并没有找到需求的内容,那么接着将会校验 PIT 中的条目。如果和 PIT 中的条目相匹配,则在 PIT 的请求接口表中添加相应的条目并删除此 Interest 包。如果后来的请求是对先前已经请求内容的请求,节点并不会因此而转发多个 Interest 包。对于相同内容的请求,一个节点只会转发一个 Interest 包。

如果在 PIT 中也没有匹配到相应的条目,那么此 Interest 包将会根据 FIB 中的条目进行转发。假定 FIB 知道所有内容的源服务器,因此它能够有效地转发 Interest 包来获取相应内容。

3.2 Data 包的处理

相比于 Interest 包,Data 包的处理过程比较简单。当 Data 包到达一个节点的接口时,Data 包的内容名将进行最长前缀匹配。如果此内容名和 CS 中的条目相匹配,说明此内容已经存储在 CS 中,那么此内容将会被丢弃。如果和 FIB 中的条目相匹配,则表明该内容名在 PIT 中没有相匹配的条目,也就是说该内容并没有被请求过,此内容也会被丢弃。如果和 PIT 中的条目形成了匹配,则表明此节点确实发送过请求此内容的 Interest 包。接着将会进行内容验证并把此内容存于 CS 中,最后依据 PIT 中匹配的条目创建一个请求接口表,并根据此表将 Data 包发送给每一个请求者。

4 CCN 相关的缓存算法

CCN 的缓存策略大致可以分为两类。一类是缓

存决策策略,即选择网络中最合适的节点去缓存相应内容。另一类是缓存替换策略,即当节点缓存空间已满时选择丢弃哪个内容来为新的内容腾出空间^[7]。

CCN 有以下三种典型的缓存决策策略:

(1) Leave Copy Everywhere (LCE)^[8]:这是 CCN 默认的缓存决策策略。其核心思想是将内容存储在沿着需求路径的每一个路由器上,但是 LCE 有较大的数据冗余度,造成缓存空间的大量浪费。

(2) Leave Copy Down (LCD)^[9]:当缓存命中时,仅在命中节点的下游节点缓存该内容,这种策略减少了需求路径上重复内容的数量,但降低了缓存的命中率。

(3) Probabilistic Caching (ProbCache)^[10]:其核心思想是节点越靠近用户其缓存概率越大,这种策略能将内容快速地缓存到网络边缘,但是并未考虑内容的流行度问题,会增加不同内容在边缘节点的竞争。

CCN 中的缓存替换策略大多使用最近最少使用置换算法(Least Recently Used, LRU)^[11]和最不经常使用置换算法(Least Frequently Used, LFU)^[12]来最大限度地存储内容。

目前,研究者们大多致力于对缓存决策策略的研究。文中主要介绍并分析以下几种具有代表性的缓存决策策略。

4.1 基于节点缓存容量的缓存策略

这种算法在缓存决策时考虑到了节点的缓存容量^[13],把它作为选择缓存节点时的一个参数。这里的缓存容量表示网络节点的 CS 中还能存储多少内容。为了记录这个缓存容量,在 Interest 包中添加了 Cache Capacity Value (CCV) 域。当节点收到一个 Interest 包后,如果在其 CS 中没有找到需求的内容,它将会把自己的缓存容量添加到 CCV 域中并转发 Interest 包。当下一个节点收到这个 Interest 包后,如果在它的 CS 中也没有找到这个内容,那么它将会把自己的 CCV 值和 Interest 包中的 CCV 值进行比较。如果它的 CCV 值小于 Interest 包中的 CCV 值,那么它会直接转发此 Interest 包。如果大于,它将会用自己的 CCV 值替换 Interest 包中的 CCV 值并转发此 Interest 包。当 Interest 包到达存有内容的服务器之后,服务器会根据最大的 CCV 值分配转发路径中相应的节点去缓存 Data 包中的内容。

这种算法在通常流量和突发流量情况下都有很好的表现。大部分缓存算法中都没有考虑突发流量的情况,但在实际情况中,突发流量是很有可能出现的。因此考虑突发流量情况是该算法的一个优点。

现在假设有六个 Interest 包,每个包请求 10 MB 的数据。当这六个 Interest 包到达一个 CCV 值是 50 MB 的节点后,认定这个节点是传输路径中拥有最大 CCV

值的节点。因此当 Data 包返回时前五个内容将会被缓存到此节点中。而当第六个 Data 包到达时,由于缓存空间已满,此时节点必须执行缓存替换策略来为第六个内容腾出缓存空间。如果这种情况大量发生,那么节点将会忙于执行替换策略,这会降低网络整体的性能。

4.2 基于节点核心值的缓存策略

文献[14]采用节点的核心值作为一个参数来选择缓存内容的节点。核心值用来衡量一个节点在通信模式中的重要性。一个节点出现在内容传输路径的次数越多,其核心值越高。

假设网络拓扑结构如图 3 所示。

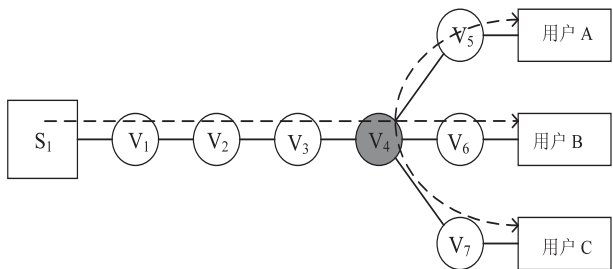


图 3 路由器的核心值

用户 A 发送一个 Interest 包,此 Interest 包将会到达服务器 S_1 ,服务器 S_1 将会沿路径 $S_1-V_1-V_2-V_3-V_4-V_5$ 发送 Data 包。根据文献[15]可知, V_4 的核心值最高。根据文献[14]中的算法,选择 V_4 作为缓存内容的节点。

这种算法选择了在传输路径中出现次数最多的节点作为缓存内容的节点。由于此节点很靠近网络边缘,因此它能够为大数目的请求提供服务。同时仅选择一个节点来缓存内容使得资源的利用率很高。

不过这种算法并没有考虑突发流量的情况,当突发流量发生时,缓存内容的节点负载会剧增,从而影响服务质量。不仅如此,该算法对于缓存节点的选择仅依据节点在拓扑中的位置而并没有考虑内容的流行度问题,因此在缓存节点中,流行的内容很可能被不流行的内容替换掉,从而使流行的内容无法被充分利用。

4.3 基于内容流行度的缓存策略

文献[16]提出一种思路,每个节点记录对每个内容的需求次数,然后以成对的形式把内容名和流行值记录到一个流行度表(Popularity Table)中。一旦一个内容的流行度达到流行度临界值之后,这个内容将会被标记为流行的。

如果某个节点拥有这个内容,它将会通过一条建议信息(Suggestion)去建议它的邻节点缓存这个内容。在发送建议信息之后该节点将会重置该内容的流行度,防止重复地将该内容发送给邻节点。MPC 缓存的示例如图 4 所示数据。

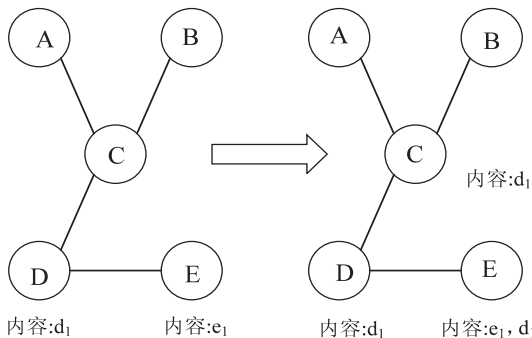


图 4 MPC 缓存示例

假设 A, B, C 节点请求内容 d_1 , A 节点请求内容 e_1 , 每次请求使得相关内容的流行度增加 1, 并假设流行度临界值为 3。此时 D 节点中内容 d_1 的流行度达到 3, 达到了临界值, 这时 D 节点将建议其邻节点 C 和 E 对 d_1 进行缓存。

这种算法依据流行度对内容进行缓存, 由于流行度高的内容被存储在了更多的节点中, 因此算法的缓存命中率相比传统的 LCE 算法有很大提升。此外, 由于缓存内容的减少(仅缓存流行度高的内容), 节约了节点的缓存空间并减少了缓存的操作次数。但也因此使得网络中内容的多样性有所下降。

4.4 核心协作缓存策略

文献[17]的主要思想是, 通过构造支配集来构建一个虚拟骨干网络。文献[18]介绍了如何构造一个连接的支配集(Connected Dominating Set, CDS)。核心节点是构成 CDS 的主要部分, 其余节点被称为常规节点。每个核心节点为一个或者几个常规节点提供服务。流行度高的内容被存储在核心节点, 因此减少了内容的冗余度。同时, 常规节点不允许缓存内容。当核心节点的缓存空间已满时, 被删除的内容将发送给其服务的常规节点。核心节点每删除一个内容将相应地增加一个条目。

算法中并不是所有节点都需要缓存内容, 大部分的运算都集中在核心节点。由于只有少部分的节点需要进行运算, 因此网络的资源利用率得到了提高。同时该算法考虑了内容的流行度问题。

构建虚拟骨干网络的过程十分复杂, 这正是该算法的瓶颈所在。如果一个核心节点失去连接, 那么与之相连的其他核心节点也会失去连接, 这将导致网络中断。

4.5 基于 Modulo hashing 的分布式协作缓存策略

文献[19]针对大视频文件, 提出了一种基于 Modulo hashing 的协同缓存策略^[20]。当某个内容块到达一个缓存节点时, 该节点将通过一个 Modulo hashing 函数计算出该内容块应该由哪个邻居节点或者自身来缓存。

如图5所示,将需求的视频内容分为很多内容块,每一个节点并不缓存全部的内容块,而是分别缓存部分内容块,由 k 个节点共同协作缓存全部的内容块。规定,每个节点都拥有一个标签,该标签是一个小于固定整数 k 的自然数。每个节点只对符合一定缓存规则的内容块进行缓存。若内容块标号对 k 取余的结果与该节点的标签相等,则节点缓存此内容块。如图所示,假设总块数为21(内容块0到内容块20)且 $k=3$,那么每个节点(路由器) r_i 的标签为 i (0,1,2)。路由器 r_0 缓存的内容为内容块 $\{0,3,6,\dots,18\}$,也就是连续10个其标号对3取余为0的内容块。

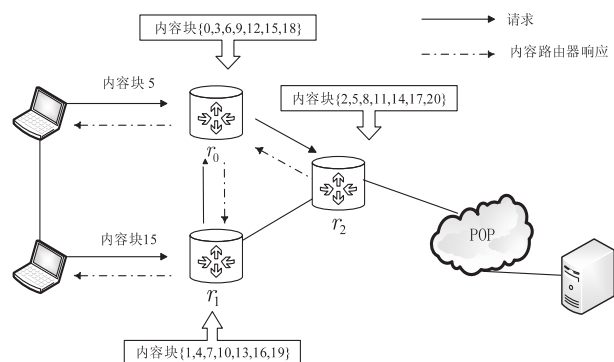


图5 基于 Modulo hashing 的协同缓存策略

这种算法使用了 Modulo hashing 使得一个节点与其 $k-1$ 个邻节点共同协作完成对内容的缓存。由于将内容切分并存储在不同的网络节点中,因此网络中内容的多样性得到了很大提升,从而许多对于内容的请求不再需要源服务器来提供服务,这有效地减少了服务器的负载。但由于这是一种基于 Modulo hashing 的算法,当网络中增加或者移除一个路由器时,先前的模值将会发生变化,之前缓存的内容的位置也必将随之更改。同时网络将会出现负载均衡的问题,使得某些路由器负载过大。

4.6 基于 Consistent hashing 的分布式协作缓存策略

这种算法是一种基于 Consistent hashing 的协同缓存算法^[21]。首先在一组路由器中根据路由器缓存容量的大小分配不同个数的 Keys,容量越大分配的 Keys 个数越多,Keys 的范围为0到 n 并由这些 Keys 组成一个 Hash 环。

如图6所示,假设路由器1拥有5个 Keys,分别是2,6,7,12,13(根据容量随机分配),路由器2拥有4个 Keys,分别是1,5,8,10,路由器3拥有0,4,9,路由器4拥有3,11。当一个内容块到达路由器时,首先计算其 hash 值并根据它的历史和当前请求频率计算出它的流行度。接着用其 hash 值和该路由器拥有的 Keys 值进行匹配,如果相匹配且其流行度大于临界值则缓存该内容,否则不缓存。例如,路由器1将只缓存 hash 值为2,6,7,12,13,且流行度大于临界值的内容块。

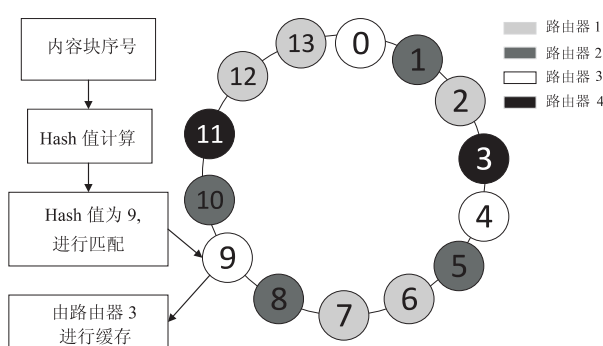


图6 基于 Consistent hashing 的协同缓存策略

这种算法考虑了路由器容量的大小问题,这在实际情况中非常值得考虑。同时算法还将流行度纳入为决策的要素,有效地提高了缓存的命中率,减少了服务器的负载。同时算法还利用 Consistent hashing 有效降低了增加或者移除路由器时造成的影响。

但 Consistent hashing 对内容块进行分布式缓存,网络边缘节点并不能够缓存流行度最高的内容,这会导致命中距离的提升。而且算法综合考虑了多方面的因素,也因此使其复杂度大大提升,这将使节点需要大量的资源去进行计算,降低了网络的整体性能。

5 结束语

随着网络规模的增长,新兴业务的不断出现,以及用户对服务的需求越来越多样化,传统的 IP 网络基础架构已经不堪重负,成为了网络进一步发展的瓶颈。为了解决这个问题,提出了很多关于未来网络方面的研究。文中涉及的 CCN 是未来网络发展方向之一。其以用户需求为中心,支持网络内所有节点的缓存,有效提高了内容的传播效率,从而克服了 IP 结构效率低的问题,使得用户能够更便捷地获取他们想要的内容。其众多优点使得它很有可能在不久的将来取代 IP 的主体地位,也因此使得它成为当前对未来网络方面研究的重点。

围绕内容中心网络重点介绍并讨论了几种具有代表性的缓存决策策略,详细分析了各自的优缺点。诸如,基于节点缓存容量的缓存策略考虑到了节点的缓存容量,提升了突发流量下的缓存性能,却容易因为频繁执行替换策略而降低了网络的整体性能;基于节点核心值的缓存策略提高了网络资源的利用率,却没有考虑内容流行度的问题;基于内容流行度的缓存策略,显著改善了缓存命中率,但也降低了网络内容的多样性;核心协作缓存策略考虑了内容流行度的同时提高了资源利用率,但核心节点失去连接时将会导致网络中断;基于 Modulo hashing 的分布式协作缓存策略提高了缓存内容的多样性,减少了服务器的负载,但增加或者移除一个路由器时会对算法造成影响;基于 Con-

sistent hashing 的分布式协作缓存策略有效降低了增加或者移除路由器时造成的影响,却使得算法的复杂度大幅提升,从而降低了网络的整体性能。综上所述,CCN 仍有诸多理论和技术问题有待解决,需要深入的研究。

参考文献:

[1] Xylomenos G, Ververidis C N, Siris V, et al. A survey of information-centric networking research [J]. IEEE Communications Surveys & Tutorials, 2014, 16(2): 1024–1049.

[2] Jacobson V, Smetters D K, Thornton J D, et al. Networking named content [C]//International conference on emerging networking experiments & technologies. [s. l.]: ACM, 2011: 1–12.

[3] 胡 骞, 武穆清, 郭 嵩. 以内容为中心的未来通信网络研究综述[J]. 电信科学, 2012, 28(9): 74–80.

[4] 夏春梅, 徐明伟. 信息中心网络研究综述[J]. 计算机科学与探索, 2013, 7(6): 481–493.

[5] Newman M. Pareto distributions and Zipf’s Law [J]. Contemporary Physics, 2004, 46(5): 323–351.

[6] 闵二龙, 陈 震, 许宏峰, 等. 内容中心网络 CCN 研究进展探析[J]. 信息安全, 2012(2): 6–10.

[7] 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.

[8] Jiang A, Bruck J. Optimal content placement for en-route web caching [C]//IEEE international symposium on network computing & applications. [s. l.]: IEEE Computer Society, 2003.

[9] Laoutaris N, Syntila S, Stavrakakis I. Meta algorithms for hierarchical web caches [C]//IEEE international conference on performance, computing, and communications. [s. l.]: IEEE, 2004: 445–452.

[10] Psaras I, Chai W K, Pavlou G. Probabilistic in-network caching for information-centric networks [C]//Proceedings of the second edition of the workshop on information-centric networ-

king. [s. l.]: ACM, 2012: 55–60.

[11] Megiddo N, Modha D S. Outperforming LRU with an adaptive replacement cache algorithm [J]. Computer, 2004, 37(4): 58–65.

[12] Petev P G, Wintergerst M. Least frequently used eviction implementation; U. S. , 7 552 284 [P]. 2009–06–23.

[13] Kim D, Lee S W, Ko Y B, et al. Cache capacity-aware content centric networking under flash crowds [J]. Journal of Network & Computer Applications, 2015, 50(C): 101–113.

[14] Chai W K, He Diliang, Psaras I, et al. Cache “less for more” in information-centric networks (extended version) [J]. Computer Communications, 2013, 36(7): 758–770.

[15] Wassermann S, Faust K. Social network analysis: methods and applications [M]. Cambridge: Cambridge University Press, 1994.

[16] Bernardini C, Silverston T, Festor O. MPC: popularity-based caching strategy for content centric networks [C]//IEEE international conference on communications. [s. l.]: IEEE, 2013: 3619–3623.

[17] Xu Y, Li Y, Lin T, et al. A dominating set-based collaborative caching with request routing in content centric networking [C]//IEEE international conference on communications. [s. l.]: IEEE, 2013: 3624–3628.

[18] Guha S, Khuller S. Approximation algorithms for connected dominating sets [J]. Algorithmica, 1998, 20(4): 374–387.

[19] Li Z, Simon G. Time-shifted TV in content centric networks: the case for cooperative in-network caching [C]//IEEE international conference on communications. [s. l.]: IEEE, 2011: 1–6.

[20] 刘外喜, 余顺争, 蔡 君, 等. ICN 中的一种协作缓存机制 [J]. 软件学报, 2013, 24(8): 1947–1962.

[21] Thar K, Oo T Z, Pham C, et al. Efficient forwarding and popularity based caching for content centric network [C]//International conference on information networking. [s. l.]: IEEE, 2015: 330–335.

(上接第 74 页)

energy consumption via rate-adaptation and sleeping [C]//Proceedings of the 5th USENIX symposium on networked systems design and implementation. [s. l.]: USENIX Association, 2008: 323–336.

[9] Heller B, Seetharaman S, Mahadevan P, et al. ElasticTree: saving energy in data center networks [C]//NSDI. [s. l.]: [s. n.], 2010: 249–264.

[10] Vasi G N, Kosti G D. Energy-aware traffic engineering [C]//International conference on energy-efficient computing and network. [s. l.]: [s. n.], 2008: 169–178.

[11] Zhang M, Yi C, Liu B, et al. GreenTE: power-aware traffic engineering [C]//IEEE international conference on network protocols. [s. l.]: IEEE Computer Society, 2010: 21–30.

[12] Tu R, Wang X, Yang Y. Energy-saving model for SDN data centers [J]. Journal of Supercomputing, 2014, 70(3): 1477–1495.

[13] Mahadevan P, Sharma P, Banerjee S, et al. A power benchmarking framework for network devices [M]//Networking. Berlin: Springer, 2009: 795–808.

[14] OpenFlow switch specification 1. 4. 0 [EB/OL]. 2013. <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf>.

[15] Ryu B, Cheney D, Braun H W. Internet flow characterization: adaptive timeout strategy and statistical modeling [C]//Workshop on passive and active measurement. [s. l.]: [s. n.], 2001.