

基于跨层信息的信息中心网络缓存算法研究^①

王伟^{②*} 孙毅* 李军* 李忠诚*

(* 中国科学院计算技术研究所 北京 100190)

(** 中国科学院大学计算机学院 北京 100190)

摘要 为了提高信息中心网络(ICN)的性能,研究了 ICN 缓存算法的优化设计。针对现有的缓存算法冗余度高、命中率较低的问题,提出了一种基于跨层设计的全新的 ICN 缓存算法 CRCache,该算法根据网络层信息与应用层信息之间的相关性,有选择地将内容缓存在与其相匹配的路由器上。通过这种跨层的缓存设计,CRCache 缓存算法能够消除网络中不必要的冗余内容、提高缓存命中率、减少网络流量。设计并实现了基于真实用户访问日志与真实中国网络拓扑的大规模仿真实验,通过实验证明,该算法在缓存命中率、网络流量减少率、服务器负载等指标上均优于现有的缓存算法。

关键词 信息中心网络(ICN),缓存算法,跨层设计,大规模仿真,内容流行度

0 引言

据 Gartner 统计,在过去的 4 年中,全球的 IP 流量增长了 4 倍,而且在未来的两年内仍然以每年高达 23% 的速度增长^[1]。互联网上呈爆发式增长的内容,如网页、高清视频、用户生成内容等,使传统的 IP 网络不堪重负。近年来,研究界提出了一种以内容(信息)为中心的称为信息中心网络(information-centric network, ICN)的革命性网络架构,来解决基于位置的 IP 网络所带来的传输效率低和扩展性差的问题。典型的 ICN 网络^[2-4]有效地提高了内容的传输效率和增强了用户体验,网络的负载也大大降低。ICN 网络性能的提升在很大程度上得益于其网络设备具备缓存的能力,而且其基于内容标签的路由算法使用户请求可以在就近的网络设备上得到响应。因而如何设计缓存算法对 ICN 网络性能的影响非常大,于是缓存策略成为热点研究方向。经典的沿路径缓存算法 LCE(leave copy everywhere),即

将用户获取的内容缓存在传输路径上的每一台路由器上,是一种有效的且易于实现的缓存算法,被广泛的用于信息中心网络(ICN)。然而基于 PPTV^[5]真实用户日志与网络拓扑的实验表明,沿路径缓存算法(LCE 算法)的 ICN 存在两个方面的缺陷:首先,跟海量的互联网内容相比,路由器的缓存容量相对来说非常小,以 PPTV 的数据为例,单在其视频点播平台上,视频内容的增长量就达到每小时约 1TB,然而典型 ICN 路由器的缓存大小只在 GB 数量级,而且该缓存资源又被很多的服务提供商共享,因此 ICN 中路由器设备的缓存资源非常稀缺;其次,这些稀缺的缓存资源并没有被高效地利用。经典的沿路径算法简单地将用户请求的内容缓存在每台路由器中,从而使沿路径的每台路由器都缓存着那些最流行的内容,冗余度很高,而且只有那些缓存在网络边缘的距离用户较近的内容才可能被访问到,其余的内容几乎不被访问。文献[6]通过大规模的仿真实验证明,以 1GB 的缓存空间为例,LCE 算法的平均缓存命中率小于 1%,网络总体流量的减少小于

① 973 计划(2012CB315806, 2012CB315802), 863 计划(2015AA016101), 江苏省未来网络前瞻性研究基金(BY2013095-1-11, BY2013095-1-13, BY2013095-1-02), 内蒙古自然科学基金(2015MS061)和国家文化科技提升计划项目基金(201201-02)资助项目。

② 男,1989 年生,博士生;研究方向:计算机网络,云数据中心网络;联系人,E-mail: wangwei2012@ict.ac.cn
(收稿日期:2015-07-14)

4%, 服务器负载的减少量小于 6%。

为了提高缓存性能, 已经研究出了新的缓存算法, 新算法大致分为两类, 一类, 称为基于应用层信息的算法, 例如根据应用层的内容的流行度来决定如何缓存内容^[7], 另一类称为基于网络层信息的算法, 该类算法根据路由器的性能及网络特性(比如带宽、中心度、流量等)等网络层的信息来决定如何缓存内容^[8]。跟经典的 LCE 相比, 这些方法确实提高了缓存利用率, 提高了网络性能。然而, 仅仅利用单一层次的信息(或者应用层或者网络层)来设计缓存算法, 无法实现最优的缓存利用率和命中率。因此, 同时考虑应用层和网络层的信息来设计 ICN 缓存算法是非常必要的。本文提出了一种采用跨层信息的 ICN 缓存放置算法——CRCache 算法, 该算法能充分挖掘网络层信息与应用层信息之间的相关性, 从而有选择性地内容缓存在与其匹配的路由器上, 大大降低了整个网络中缓存的内容的冗余度, 提高了内容的多样性, 从而降低整个网络的负载同时提高了缓存命中率与利用率。为了验证本文提出的算法的性能, 我们基于 PPTV 公司的真实用户日志与真实的网络拓扑, 进行了大规模的网络仿真实验。实验表明, 与现有的单层算法相比, CRCache 算法将缓存命中率提高了 62.5%, 减少了至少 42% 的网络流量, 服务器负载和用户体验都得到优化。

1 ICN 缓存及相关工作

1.1 ICN 缓存

信息中心网络(ICN)缓存与传统的 Web 缓存、CDN 缓存有着明显的区别, 主要表现在三个方面^[9]: 缓存透明化、缓存泛在化、缓存对象细粒度化。首先, ICN 中为每个内容对象分配一个全局唯一的标识, 并且根据内容标识计算路由、转发与响应用户请求。因此网络中缓存的内容可以被任意的上层应用所使用, 只要缓存的内容与请求内容标识一致。其次, 传统的 CDN 缓存或 Web 缓存有确定的缓存服务器, 只需要对内容的访问频率、流量分布进行分析便可获得最优化的缓存策略。然而在 ICN 中, 全网 ICN 路由器都具有缓存的能力, 而且拓扑

结构复杂, 呈现网状而非传统的层次化结构, 因此也给缓存放置策略提出了新的挑战。最后, ICN 中通常以比文件粒度更小的数据块(chunk)作为缓存单元, 因此对 chunk 的特性如流行度、空间大小等需要全新的刻画来满足缓存策略的要求。经典的 LCE 算法并没有很好地利用这些新的特征。

1.2 相关工作介绍

当前已经有一些研究成果根据 ICN 的特性来提高 ICN 的缓存性能, 本小节我们简单地介绍这些相关的工作, 并分析其优缺点。

正如在引言中所提出的, 现有的缓存算法中有一类算法通过关注应用层信息, 即内容本身的特性, 来决定缓存策略。在文献[10]中, Laoutaris 等提出了一种叫做 LCD (leave copy down) 的算法, 该缓存算法被用在大规模的层次化的 Web 系统中。具体来说, 每当一个用户请求在某个设备(服务器或中间路由器)被响应, 那么该内容就被缓存在较低一级的设备中, 相当于该内容距离终端用户更近了一跳。LCD 算法的本质是将用户访问量较大的内容一跳一跳地推向用户。文献[7]利用形式化的方法根据内容的流行度进行建模, 以一种合作式的方式来决定传输路径上的缓存位置。在文献[11]中, Ming 等提出了一种基于老化的合作式的缓存算法, 该算法利用一种轻量级的合作机制将最流行的内容缓存在网络边缘, 同时充分利用剩余的中间路由器缓存资源来存储其他内容。然而将最流行的内容缓存在网络边缘并不是最优的方法, 因为网络边缘的路由器接入的用户较少, 对于直接接入的用户可能效果较明显, 而总体来说缓存在网络边缘的流行内容被访问的次数并不高。这类方法重点关注内容本身的特征, 主要根据内容的流行度来决定缓存决策, 在一定程度上提高了缓存性能, 但是并没有结合网络的特性来最优化流行内容的冗余度及缓存的位置, 缓存效率较低。

还有一类工作提出了利用网络层的信息来优化缓存策略。文献[8]考察了多个基于图的节点中心度的指标(例如 betweenness、closeness、stress 等)来为 ICN 网络设备分配缓存空间。该文指出, 根据节点中心度分配缓存空间的方法有效提高了网络缓存

性能并且节约了缓存资源。文献[8]也提出可以在网络中选择中心度较大的节点上进行缓存,可以提高网络缓存的整体性能。这类方法实现较为简单,部署容易,但是只考虑了网络的拓扑特征,无法将流行内容和非流行内容最优化地缓存在最合适的路由器中。

此外,文献[12-14]使用了概率模型来设计缓存算法。在文献[12]中,Psaras 等提出了一种叫做 ProbCache 的方法,该方法通过对路径上的路由器的缓存能力进行建模,提出一种基于概率的算法来充分利用缓存资源,降低内容冗余度。文献[13,14]使用了一种基于随机数的概率方法来决定内容的缓存位置。

然而,在现有文献中,几乎没有见到充分挖掘网络层信息与应用层信息之间的紧密关联关系,并且基于此来设计缓存算法的相关工作。仅仅关注单一层面的信息并不足以最优化缓存算法。因此,本文基于以上动机与启发,设计了一种跨应用层和网络层信息的缓存算法来提高 ICN 网络的缓存性能。

2 采用跨层信息的 ICN 缓存算法设计

2.1 CRCache 算法概况

为了充分利用 ICN 网络的缓存资源和提高缓存命中率,本文提出了一种基于应用层和网络层信息的缓存放置算法(CRCache)。根据帕累托法则对 PPTV 大量数据的分析表明,网络中 90% 以上的用户请求集中在少于 20% 的热点内容上,如图 1 所示,我们将 PPTV 日志中的视频内容按照访问频次进行统计,计算其累积分布函数(CDF)值,发现内容的访问频次遵循 Zipf 分布。其次,根据对真实网络的分析^[6],计算出各路由器通过的流量值,如图 2 所示,我们可以发现网络中存在一些连接度高的且实时流量很大的路由器,这些路由器被称为热点路由器,而其他的路由器则流量较小。基于以上两点分析,如果将访问量大的热点内容有选择性地缓存在少数几个热点路由器中,那么将会大大提升缓存命中率,同时将次热的内容缓存在次热的路由器中,即利用内容热度与路由器热度的匹配关系来决定内

容的缓存位置,将会显著降低缓存内容的冗余度,同时提高缓存利用率。

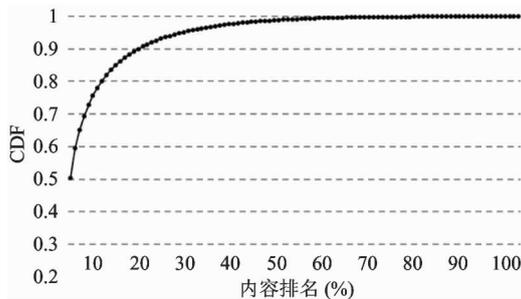


图 1 内容访问频度分布

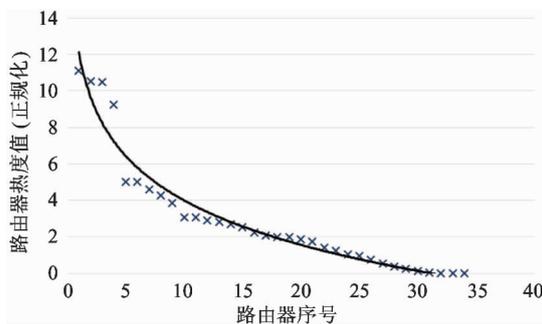


图 2 路由器热度分布

CRCache 算法旨在通过充分挖掘应用层内容的信息(如内容类型、流行度)与网络层路由器的信息(如度数、连接性、实时流量)之间的匹配程度并基于此来决定内容的缓存位置,如图 3 所示,因而提高了网络中热点内容的命中率,降低了网络中缓存内容的冗余度,同时网络中内容的多样性也大大增加,提高了缓存空间的利用率。

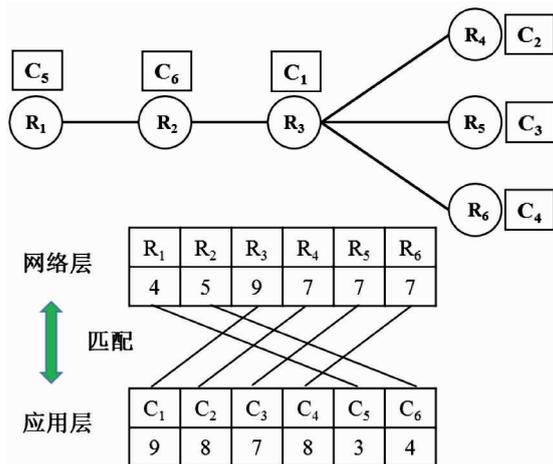


图 3 CRCache 算法缓存策略示意图

假设路由器($R_1 - R_6$)缓存空间为 1,路由器中心度与内容热度如图中的表所示。根据匹配程度,CRCache 算法将最热的内容 C_1 缓存在中心度最高的路由器 R_1 中,次热的内容 C_2, C_3, C_4 缓存在边缘路由器中,剩余内容分别缓存在中心度较低的路由器中。

具体来说,CRCache 算法首先需要定义应用层的内容特征信息和网络层的路由器特征信息并计算其特征值。由于内容特征信息(如内容的流行度)与路由器特征信息(如实时流量)在网络中实时变化,我们还需要动态地更新与维护这些信息,这部分内容将在 2.2 和 2.3 节中详细介绍。基于这两种特征信息的定义,我们在 2.4 节中提出了一种缓存放置算法,该算法用来确定内容在传输路径上的缓存位置。具体来说,当用户请求的内容到达某路由器时,我们首先获取其内容信息特征值,并计算其与本地路由器特征值之间的匹配值,并以此值决定是否放置在缓存中(或替换缓存中的内容)。

2.2 应用层特征信息设计空间

ICN 缓存策略需要准确地对缓存内容的特征信息(如流行度、类型等)进行刻画,这将有利于提高缓存命中率,提高缓存整体性能。由于 ICN 缓存所具有的泛在化、细粒度的新特性,传统的对缓存内容的刻画已不适用,因此我们提出了一种全新的 ICN 缓存内容的特征信息的描述方法。

为了准确地体现 ICN 缓存对象的特征信息,我们从缓存对象类型(T)、缓存对象流行度(P)、缓存对象前后关联性(R)三个方面对缓存对象进行刻画。

缓存对象类型(T):网络中会承载各种应用程序的数据,但是并不是每种类型的数据对象都需要被缓存,如即时消息类应用,社交网络消息。而缓存视频类、Web 类、下载类的内容将会明显提高缓存性能与用户体验,因为这些内容文件块较大,且更新频率小。根据思科的数据报告^[1],截至 2013 年,视频类流量占全球互联网总流量的 66%,并将在 2018 年达到 79%。因此,我们首先需要根据应用类型判断是否需要缓存,再根据其所占有的网络流量决定各类应用的优先级。根据不同的需求设置,我们可

以对不同的应用程序的数据赋予不同的优先级,我们根据思科和中国互联网报告^[15],给出一种的类型优先级值(T)的设置方法,如表 1(值越高优先级越高)所示。

表 1 缓存对象类型优先级

应用类型	T 值
视频音乐类	0.8
网页类	0.1
文件下载类	0.09
其他	0.01

缓存对象流行度(P):传统的缓存系统一般以文件作为缓存单元,而 ICN 中由于其线速工作的要求^[16],通常以粒度更小的数据块(chunk)作为缓存单元。因此,我们以 chunk 为缓存单元重新定义了缓存对象的流行度。

缓存对象的流行度会随着时间的变化而变化,因此我们设计了一种动态的流行度计算方法。我们的方法将对象的流行度分为两种情况:初始流行度(BP)和动态流行度(DP)。

当一个新的内容被添加到内容服务器中时,该内容会被分配一个初始化的流行度(BP)。初始流行度可以根据内容的自身属性(metadata,如导演、演员、影片分类等)和内容提供商的先验经验进行设置,此过程用 S 表示。其次,相同的视频文件中不同的 chunk 的流行度也是不同的。如图 4 所示,我们根据 PPTV 数据,对同一个视频文件不同时间段的流行度进行分析,得到其 chunk 级别的流行度分布(不同应用程序的 chunk 流行度可根据同理获得)。我们可以发现,在视频开始的前 10min 的访问频次非常高,视频的中间部分访问频次比较稳定,而最后的几分钟又迅速下降。我们可以基于此拟合出同内容不同 chunk 的流行度分布函数(Z)。因此根据 S 和 Z ,我们便可得到 chunk c 的初始流行度

$$BP_c = S(c.metadata) + Z(c) \quad (1)$$

初始流行度只作为缓存系统中对新增加内容的流行度的预测,之后该内容的流行度会根据网络中的访问频率、上线时间等进行动态更新。

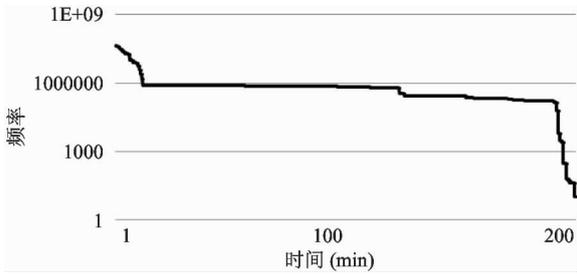


图4 同一个文件的不同 chunk 的流行度分布

当内容被缓存在 ICN 路由器时,路由器的缓存会为每个缓存的内容维护一个动态流行度(DP),并以该内容携带的流行度作为初始值。内容的动态流行度根据路由器本地的访问频率等进行更新,因此在网络中,相同的内容在不同的路由器中维护的动态流行度也不相同。动态流行度的更新包括访问频率和“老化”机制两个参数,如下式:

$$DP_c = DP_c + \frac{HitTime}{\Delta T} - Aging_c(T_1, T_2) \quad (2)$$

根据式(2),我们每隔 $\Delta T = T_2 - T_1$ 的时间更新一次本地路由器缓存中的内容的流行度,更新的值包括两项,第一项 $\frac{HitTime}{\Delta T}$ 代表在 ΔT 时间内该内容的命中次数,反映出在上个时间段内的访问频率;第二项 $Aging_c(T_1, T_2)$ 表征缓存对象的老化机制。内容流行度的老化机制是指随着时间的变化,内容的流行度也会衰减。衰减的速度与时间、内容的历史流行度相关,定义如下:

$$Aging_c(T_1, T_2) = \lambda_c(T_1, T_2) \times (T_2 - T_1) \quad (3)$$

其中 $\lambda_c(T_1, T_2)$ 表示内容 c 在 (T_1, T_2) 时间段内的衰减速率, $\lambda_c(T_1, T_2)$ 等于在过去的 N 天内,内容 c 在 (T_1, T_2) 时间段内的平均访问次数。因此 $Aging_c(T_1, T_2)$ 的物理含义是内容 c 在 (T_1, T_2) 时间段内的期望访问次数。因此,使用该机制可以使内容的动态流行度随着访问频率的相对值的增加而增加,同时随着访问频率的相对值的下降而下降。这也非常符合我们对 PPTV 数据的观测,即一个热点内容的流行度会快速增长,同时也会快速下降。

缓存对象前后关联性(R):根据以往研究表明^[17],前后相连的 chunk 在访问频率上有很强的关联关系。在我们对内容进行刻画时也考虑了 chunk 之间的关联性,当一个 chunk c 的前后相连的 chunk

的流行度较高时,该 chunk 在缓存时将赋予更高的优先级, R_c 定义为前后相连的 chunk 的平均流行度。

根据以上对缓存对象在三个关键属性的刻画,我们可以精确地定义出一个缓存对象的特征信息。根据设计需求给出这三个属性不同的权重 $[\gamma, \delta, \tau]$,从而计算得出关于缓存对象的特征值(A):

$$A_c = [T_c, P_c, R_c] \times [\gamma, \delta, \tau]^T \quad (4)$$

按 CRCache 算法,内容的特征值将与路由器的特征值结合起来决定内容缓存放置策略和替换策略。

2.3 网络层特征信息设计空间

网络层特征信息是指 ICN 路由器在网络拓扑层面的连接度、核心或边缘、实时流量等属性。由于网络中不同的路由器的特征信息不尽相同,因而忽略这些特征化的信息而采用全局统一的缓存策略很难使缓存性能得到优化。因此,我们提出将内容信息与路由器信息结合起来决定放置算法,本小节提出网络层特征信息的设计空间与方法。

本文从两方面计算路由器的特征信息:静态中心度($centrality$)和动态流量值(F)。中心度是一个非常重要的网络拓扑指标,其表征一个路由器节点(v)在网络中的连接程度,定义如下^[8]:

$$centrality(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \quad (5)$$

式中, σ_{st} 是所有从源节点 s 到目的节点 t 的最短路径的数量, $\sigma_{st}(v)$ 是这些 s 到 t 的最短路径中经过节点 v 的数量。一个路由器节点在网络中的中心度表达了在全网所有最短路径中,包含该节点的最短路径的比例,因此直接表征了该节点在网络中的重要性程度和连接程度。由于网络拓扑的变化并不频繁,我们可以提前计算好每个节点的中心度,并保存在各自的路由器中,定时更新即可。

现代的路由器都支持流量统计,每台路由器实时地统计 ΔT 时间内的流量 $F_{\Delta T}(v)$,来表征该路由器的动态流量值。结合以上两信息及相应的权重 (α),我们可以得到路由器的如下特征信息值:

$$N_v = [centrality(v), F_{\Delta T}(v)] \times [\alpha, 1 - \alpha]^T \quad (6)$$

至此,我们介绍了应用层信息和网络层信息的设计空间和基本的计算方法。需要说明的是,对于这两层信息的计算方法可以在我们提出的设计空间内重新设计来适应不同的缓存需求与场景。

2.4 基于跨层信息的缓存放置算法

为了降低 ICN 缓存的冗余度,提高缓存命中率与缓存性能,CRCache 算法结合应用层的内容特征信息和网络层的路由器特征信息来设计缓存放置算法,即将缓存对象放置在与之效用值(U)较高的路由器中。因此,效用值的计算是缓存算法的关键。

本文对效用值的计算基于以下分析。在网络中,绝大多数的用户请求集中在流行内容上。因此在网络边缘节点缓存最流行的内容可以大大提升缓存性能,提高用户体验。然而,这种做法有以下缺陷:(1)网络边缘的路由器冗余地缓存了大量相同的流行内容,从而浪费了稀缺的缓存资源;(2)边缘路由器的中心度较小,服务的用户数量少,因此对流行内容的访问量较少,也相对分散。假如我们将流行内容(特征值高的内容)缓存的位置从网络边缘向网络核心方向后退 1、2 跳,缓存在那些特征值高(中心度大,流量大)的核心路由器上,就可以解决以上提出的两个问题。由于边缘路由器的数据包排队时间较少,增加的 1、2 跳延迟对于用户的体验差异几乎没有。同时,将特征值相对较低的内容缓存在特征值相对较低的路由器中,形成对应的匹配关系,就可以大大提高网络缓存内容的多样性,同时提高缓存性能。

由于进行了以上分析,我们对效用值 U 的计算简单而高效。效用值定义如下:

$$AL_c \stackrel{Normalize}{\leftarrow} A_c, NL_v \stackrel{Normalize}{\leftarrow} N_v (AL_c, NL_v \in (0,1))$$

$$U = 1 - |AL_c - NL_v| \quad (7)$$

根据此公式,我们首先将内容 c 的特征值和路由器 v 的特征值做正规化处理,使其值在相同的数量级。 AL_c 和 NL_v 的差的绝对值越小,则其匹配程度越高,因此效用值 U 的值也越高,效用值越高表明如果将内容缓存在此路由器中将取得更高缓存性能。

根据以上分析,我们设计了缓存算法(如算法 1)。该算法将通过在各路由器中独自运行,来决策每个通过的内容是否缓存在本路由器中,算法复杂

度为 $O(N)$, N 为当前路由器中缓存内容的数量。若采用优先队列的数据结构,复杂度可以降到 $O(1)$,也符合 ICN 网络缓存的线性复杂度的要求。按算法 1,第一步,本地路由器根据式(4)每间隔 ΔT 的时间更新本地缓存中所有内容的特征值 A ,并且根据式(6)更新本地路由器的特征值 N ,利用内容和路由器的特征值根据式(7)更新每个内容的效用值 U ,效用值越大优先级越高。需要注意的是 ΔT 值的设置,由于内容的热度变化不会过快,因此频率为 0.5~1 小时即可。第二步,每当有新内容到来时,如果缓存还没有占满,则直接缓存该内容,否则第三步将根据式(4)(6)(7)计算新内容与本路由器的效用值,如果该值大于现有缓存的内容中最小的效用值,则将新内容替换掉效用值最小的旧内容(第四步)。

算法 1. 基于跨层信息的缓存放置算法

输入:缓存内容 c

输出:缓存决策(是否在本路由器缓存内容 c)

第一步:每隔 ΔT 的时间,路由器根据公式(4),更新本地缓存中的内容的特征值,根据公式(6),更新路由器特征值,并根据公式(7),更新每个内容的效用值。

第二步:如果此时缓存空间不满,将内容 c 缓存在本路由器中,退出。

第三步:如果此时缓存空间已满,则根据公式(4)(6)(7)计算内容 c 的特征值及内容 c 在本路由器的效用值 U_c 。

第四步:如果 U_c 大于缓存中所有内容的最小的效用值,则删除该效用值最小的内容,并将内容 c 缓存在本路由器中。

3 实验结果与分析

为了验证 CRCache 算法在大规模网络中的缓存性能,我们基于真实的网络拓扑和真实的用户访问日志进行了 ICN 的仿真实验,在 3.1 节介绍了使用的 ICN 仿真器、实验拓扑信息及访问日志信息,在 3.2 节介绍了实验结果与分析。

3.1 实验环境部署

网络拓扑:本实验在 PPTV 客户端软件中添加了 traceroute 插件,该插件随机地向其他 PPTV 客户

端发送 traceroute 命令,根据所有插件返回的 traceroute 信息,我们得到了一张真实的路由器级的网络拓扑。本实验中的拓扑信息基于 PPTV 的 168 万活跃用户持续两个月的 traceroute 的信息绘制而成。该网络拓扑包含 8000 多台独立路由器(跨越了 89 个网络服务提供商(ISP),6583 个 PoPs,覆盖了中国大陆 31 个省,平均路径长度为 8.26 跳,拓扑直径为 22),还有 200 台 PPTV 服务器和 450 万客户端。

访问日志:本实验的访问日志来自 PPTV 服务器的真实用户访问记录。每条用户记录包含一次访问请求,记录的信息包括用户信息(IP 地址、服务提供商、地理位置等),服务器信息(服务器 IP 地址,地理位置),以及请求的内容的信息(内容地址,观看时长,文件大小等)。本仿真实验中涉及了 27 万个视频,2600 万次的访问记录。通过统计,本实验中的用户访问日志符合 Zipf 分布,其中参数 α 的值为 1.174。

仿真平台:现有的 ICN 仿真软件,如 CCNx^[18], ndnSIM^[19]等,在大规模网络拓扑和海量用户日志下性能较差,运行时间长。因此,本实验设计了适合大规模网络仿真的 ICN 仿真平台。该仿真平台实现了 ICN 的主要特性,包括缓存、转发、路由。其次,该仿真平台运行效率高,针对亿级的用户日志,仿真实验可以在 5 个小时以内完成。同时,支持自定义的缓存放置算法、缓存替换算法、路由策略。

实验参数:本文在刻画内容特征值(式(4))和路由器特征值(式(6))时采用了权重参数,具体的设置可根据服务提供商的内容特征和网络拓扑特征进行设置,在本实验中,内容流行度的三个维度赋予相同的权重,即 γ, δ, τ 均为 $1/3$,路由器的特征值我们赋予实时的路由器流量更大的权重,即 α 设置为 0.2。另外,内容效用值的更新间隔 ΔT 为 0.5h。在实验中,我们设置了 4 种不同的缓存大小,分别为 100MB, 1G, 10G, 100G。

3.2 实验结果分析

本实验将 CRCache 算法与现有算法进行了比较,包括典型的基于应用层的缓存策略(LCD)^[10]、典型的基于网络层的缓存策略(Centrality)^[8]、基于概率的缓存策略(Fixed Probability, Fix(P))^[12]、

ICN 原生缓存策略(LCE)^[2],这些算法都使用最不经常使用的置换算法(LFU, least frequently used)作为缓存替换算法,对比的性能指标包括缓存命中率、网络流量减少率、节约跳数、服务器负载和用户体验。

实验 1 缓存命中率。缓存命中率是衡量缓存性能的重要指标。首先,我们统计了在不同缓存空间大小下(100MB, 1G, 10G, 100G)的网络缓存命中率,如图 5 所示。从图 5 中可以看出,当缓存空间只有 100MB 时,全网的缓存命中率只有不到 1%,这是由于缓存中内容非常少,而且替换次数频繁,缓存中的内容不稳定,导致用户请求很难命中缓存中的内容。随着缓存空间的增大,各算法的缓存命中率也随之增大,CRCache 算法的命中率相比其他 4 种算法有明显提升,例如,当缓存空间达到 100G 时,CRCache 算法的缓存命中率达到 23%,是 LCE 算法的近两倍。

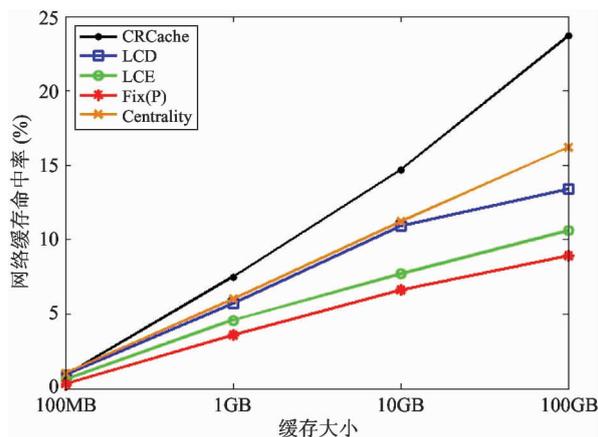


图 5 在不同缓存空间大小下的缓存命中率

其次,本实验将内容的特征值分为 10 个不同等级,然后分别考察其命中率,如图 6 所示。从图 6 中的数据我们发现,特征值等级较低的内容的命中率较低(如图中红色标注的数据),几乎都小于 1%。随着特征值的增大,命中率也相应地提高(如图 6 中绿色标出的数据)。实验发现,CRCache 算法对于高特征值的内容的命中率提升非常明显而且内容覆盖范围广,从等级 7 至等级 10 的内容,命中率从 26.5% 到 69.9%,而反观其他算法,只有等级 10 的内容命中率较高。这说明了 CRCache 算法在提高

命中率的同时,增大了缓存中流行内容的多样性,降低了内容冗余度,使更多的热点内容的命中率得到

提高。

	1	2	3	4	5	6	7	8	9	10
CRCache	0.004	0.012	0.029	0.037	0.089	0.153	0.265	0.651	0.699	0.695
Centrality	0.017	0.046	0.071	0.104	0.153	0.157	0.185	0.268	0.274	0.264
LCD	0.022	0.062	0.084	0.072	0.124	0.203	0.260	0.345	0.413	0.440
LCE	0.017	0.048	0.074	0.108	0.158	0.172	0.210	0.273	0.249	0.264
FixP	0.021	0.063	0.091	0.102	0.156	0.182	0.174	0.130	0.124	0.126

图 6 不同内容特征值下的缓存命中率

实验 2 网络流量减少率。在 ICN 中,用户请求如果在路由器命中,则将内容立刻返回,这样就大大减少了整体网络的流量,从而有效降低了网络的负载。本实验将每种算法在不同缓存空间大小下的网络流量与没有缓存的方案相比,分别得到其网络流量的减少率,如图 7 所示。从图 7 中可以看出,CRCache 在不同缓存大小下与其他方案相比都有明显提升,最好情况下,流量减少率达到 15.8%,与 LCE 方案相比提升达到 2 倍。其次,还可以看出,基于中心度的 Centrality 方案是其他四种方案中最好的,也证明了利用简单的中心度算法可以明显提升缓存性能。

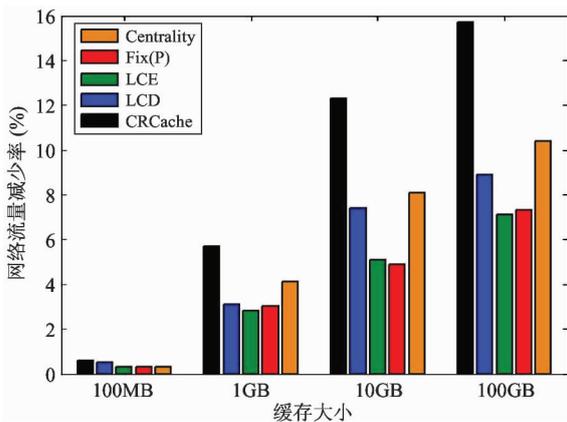


图 7 在不同缓存空间大小下的流量减少率

实验 3 节约跳数。对于所有在网络中命中的用户请求,我们更进一步地考察其节约的跳数。我们分别计算在所有命中的用户请求中,各节约跳数所占的比例,如图 8 所示。我们可以发现,LCD、LCE、

Fix(P)算法仅仅节约 1 跳的比例就分别占总命中数的 45%、36%、32%,节约 3 跳以上的比例只占 37%、46%、47%,因此对于网络负载的缓解作用并不明显。而 CRCache 算法,节约 1、2、3 跳的比例分别都只在 10% 左右,而节约 3 跳以上的比例达到近 70%。因此,通过本实验更进一步地说明,CRCache 算法利用跨层的缓存策略方法不仅总体的命中率得到提高,而且在命中的这些请求中,节约了更多的传输跳数,从而用户体验的提升更为明显。

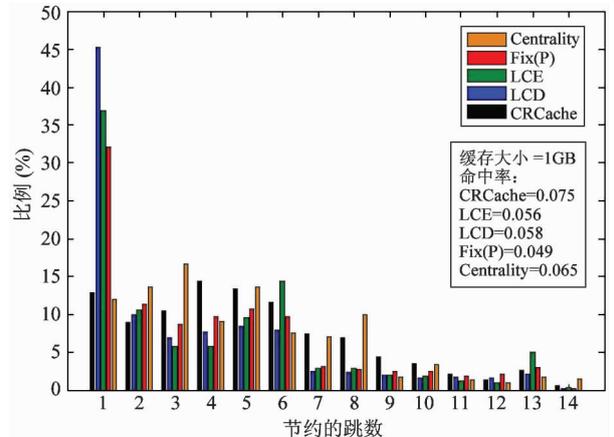


图 8 节约的跳数的比例

实验 4 服务器负载。本实验仿真了服务提供商的服务器负载,作为衡量缓存策略在服务端性能优化的客观评价指标。本实验分别在四种缓存空间参数下,统计了所有 200 台服务器接收到的请求数量并求其平均值,作为服务器负载值,如图 9 所示。可以发现,随着缓存空间的增大,5 种缓存策略的服务

器负载均明显下降。CRCache 算法将服务器负载从 91% 下降到 50%, 降幅明显高于其他四种算法。服务器负载的下降正是由于大量的用户请求在网络中得到响应, 与实验 1 结合来看, 也可以发现服务器负载与网络命中率呈正相关。

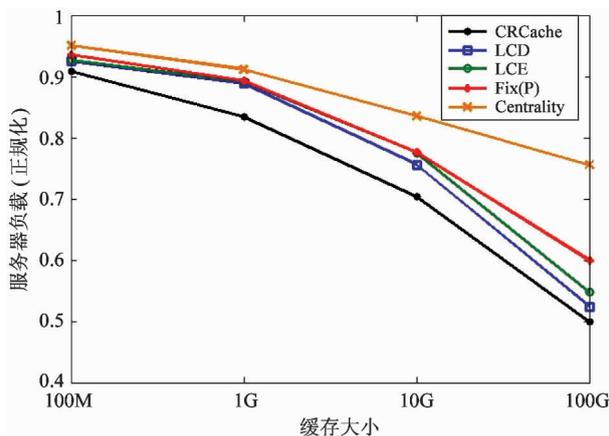


图 9 不同缓存空间大小下的服务器负载

实验 5 用户体验。本实验从用户体验的角度来评测 CRCache 缓存算法的性能。对于用户观看视频的体验而言, 最关心的是观看的流畅度, 也就是视频缓冲的次数, 次数越少体验越好。因此, 根据用户日志中的观看时间及视频码率, 以及实验中的用户的实际下载速度, 计算出平均的缓冲次数, 如图 10 所示。从图 10 中可以看出, CRCache 算法在各缓存大小下, 视频的缓冲次数相较于传统的 LCE 算法有明显下降, 最高比 LCE 算法降低了缓冲次数达 30%

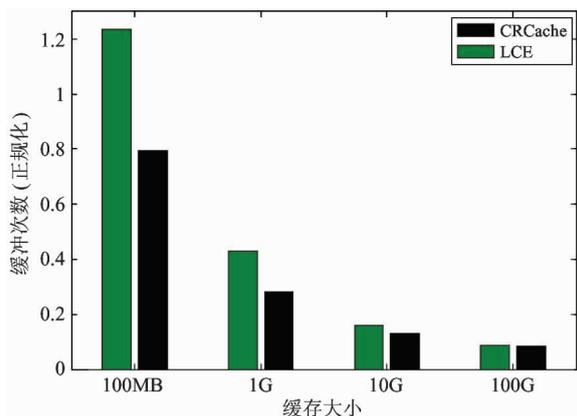


图 10 不同缓存空间下的视频缓冲次数

以上。因此, 从用户体验来讲, CRCache 算法增大了内容的下载速度, 减少了用户在观看视频时的卡顿次数, 使用户具有更好的观看体验。

通过上一系列的仿真实验, 本文分别从网络的角度(命中率、流量减少率、节约的条数)、服务端的角度(服务器负载)和用户角度(视频缓冲次数)全面衡量了 CRCache 算法的缓存性能, 相较于现有的算法, 各指标都得到明显提升。

4 结论

为了降低 ICN 网络缓存的冗余度, 提高缓存性能, 本文提出了一种基于跨层信息的 ICN 缓存算法。该算法从多个角度充分挖掘缓存内容的特征信息和路由器的特征信息之间的匹配性, 根据内容和路由器的匹配程度决定内容的缓存位置。该算法的优势在于降低了网络缓存内容的冗余度, 使全网缓存更多的热点内容, 提高了全网的缓存命中率、明显减小了网络流量、节约了更多的内容传输跳数, 内容服务器的负载和用户体验也得到优化。根据基于真实拓扑和用户日志的大规模仿真实验, 证明了 CRCache 算法在各指标均优于现有的经典算法。

参考文献

- [1] Cisco, VNI: Forecast and methodology 2013 ~ 2018. http://www.cisco.com/web/CN/assets/pdf/VNI_forecast_whitepaper_cn.pdf, 2015
- [2] Jacobson V, Smetters D K, Thornton J D, et al. Networking named content. In: Proceedings of ACM International Conference on Emerging Networking Experiments and Technology, Rome, Italy, 2009. 1-12
- [3] Jacobson V, Smetters D K, Thornton J D, et al. NDN project. <http://nameddata.net/techreport/TR001ndnproj.pdf>, 2010
- [4] Koponen T, Chawla M, Chun B G, et al. A data-oriented (and beyond) network architecture. *Acm Sigcomm Computer Communication Review*, 2007, 37(4): 181-192
- [5] PPTV. <http://www.pptv.com>, 2015
- [6] Sun Y, Fayaz S K, Guo Y, et al. Trace-driven analysis of ICN caching algorithms on video-on-demand workloads. In: Proceedings of ACM International Conference on Emerging Networking Experiments and Technology,

- Sydney, Australia, 2014. 363-374
- [7] Li J, Wu H, Liu B, et al. Popularity-driven coordinated caching in named data networking. In: Proceedings of the 8th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, Austin, USA, 2012. 15-26
- [8] He D, Pavlou G, Chai W K, et al. Cache less for more in information-centric networks. In: Proceedings of the Ifip-tc6 Networking Conference, 2012, 36(7):27-40
- [9] 张国强, 李杨, 林涛等. 信息中心网络中的内置缓存技术研究. 软件学报, 2014, 25(1): 154-175
- [10] Laoutaris N, Syntila S, Stavrakakis I. Meta algorithms for hierarchical web caches. In: Proceedings of IEEE International Conference on Performance Computing & Communications, Phoenix, USA, 2004. 445-452
- [11] Ming Z, Xu M, Wang D. Age-based cooperative caching in information-centric networking. In: Proceedings of IEEE International Conference of Computer Communication and Networks, Orlando, USA, 2014. 1-8
- [12] Psaras I, Chai W K, Pavlou G. Probabilistic in-network caching for information-centric networks. In: Proceedings of ACM Special Interest Group on Data Communication (SIGCOMM) Workshop on Information-Centric Networking, Helsinki, Finland, 2012. 55-60
- [13] Arianfar S, Nikander P, Ott J. Packet-level caching for information-centric networking: [Technology Report]. Finnish ICT-SHOK Future Internet Project, 2010
- [14] Psounis K, Prabhakar B. A randomized web-cache replacement scheme. In: Proceedings of IEEE International Conference of Computer Communication and Networks, Anchorage, USA, 2001. 1407-1415
- [15] 中国互联网网络中心. 中国互联网报告, <https://www.cnnic.net.cn/hlwfzyj/hlwzxbg/hlwtjbg/201403/P020140305346585959798.pdf>, 2014
- [16] Perino D, Varvello M. A reality check for content centric networking. In: Proceedings of the ACM Special Interest Group on Data Communication(SIGCOMM) Workshop on Information-Centric Networking, Toronto, Canada, 2011. 44-49
- [17] Cho K, Lee M, Park K, et al. WAVE: popularity-based and collaborative in-network caching for contentoriented networks. In: Proceedings of the IEEE International Conference of Computer Communication and Networks Workshop on NOMEN, Orlando, USA, 2012. 316-321
- [18] CCNx Project. <http://www.ccnx.org>
- [19] Afanasyev A, Moiseenko I, Zhang L. ndnSIM: NDN simulator for NS-3. <http://nameddata.net/techreport/TR005-ndnsim.pdf>, 2012

Design of an ICN caching algorithm based on cross-layer information

Wang Wei^{* **}, Sun Yi^{*}, Li Jun^{*}, Li Zhongcheng^{*}

(* Institute of Computing Technology, Chinese Academy of Science, Beijing 100190)

(** University of Chinese Academy of Science, Beijing 100190)

Abstract

To improve the performance of information-centric networks (ICN), the optimal design of ICN caching algorithms was studied. In view of the fact that current ICN caching algorithms have the drawbacks of high content redundancy and low cache hit rate, a novel ICN caching algorithm based on cross-layer design, called CRCache, was proposed. The CRCache algorithm caches each specific content on its matchable router according to the correlation of the network layer's information and the application layer's information. The cross-layer design can eliminate unnecessary redundant contents in ICN, improve the cache hit rate and reduce the network traffic. A large scale and real traces-driven simulation with an underlying real Internet topology in China was designed and conducted, and the results show that the caching performance of the CRCache algorithm exceeds current schemes in the respects of hit rate, traffic reduction and server load.

Key words: information centric network (ICN), caching algorithm, cross-layer design, real-trace simulation, content popularity