

一种内容中心网络中的热区控制及 内容调度缓存算法

方馨蔚,陈庶樵,江逸茗,任泽荣

(国家数字交换系统工程技术研究中心,河南郑州 450002)

摘 要: 针对内容中心网络中同质化缓存造成的节点负载不均和存储资源无法有效利用的问题,提出一种热区控制及内容调度缓存算法. 在内容请求时,根据节点介数与节点访问度综合判断节点热度,筛选出负载过重节点;缓存决策时,以流行度为依据将内容调度至空闲邻居节点,并设置生命期控制机制,从而达到分散请求、消除热区的目的. 仿真结果表明该算法减少了请求时延与路由跳数,在提高缓存命中率的同时,有效改善了负载在节点上分布的均衡性.

关键词: 内容中心网络; 协作缓存; 负载均衡; 内容调度

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2017)05-1182-07

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2017.05.022

The Hotspot Control and Content Dispatch Caching Algorithm in Content-Centric Networking

FANG Xin-wei, CHEN Shu-qiao, JIANG Yi-ming, REN Ze-rong

(National Digital Switching System Engineering & Technological Research Center, Zhengzhou, Henan 450002, China)

Abstract: A hotspot control and content dispatch caching algorithm is proposed to improve the homogeneous caching caused load imbalance and low storage utilization. When the content request arrives, the overloaded nodes are selected synthetically by betweenness and access degree. The content is dispatched to idle neighbors with popularity for caching decisions, combined with a lifetime control mechanism. So that the request balancing and the hotspot relieving are achieved. Simulation results show that, with the decrease of the request latency and hop counts, our proposed algorithm can effectively improve the load imbalance and increase the hit rate as well.

Key words: content-centric networking; collaborative caching; load balancing; content dispatch

1 引言

互联网飞速发展带来了网络用户的爆炸式增长以及数据内容的急剧膨胀,传统的端到端通信的 TCP/IP 网络暴露出越来越多的问题. 以用户兴趣为导向,信息对象本身为核心标识的内容中心网络(Content-Centric Networking, CCN)^[1,2]应运而生. 作为一种革命式的新型网络结构,CCN 以信息名字取代 IP 地址作为网络传输标识,将关注点由内容的位置(where)转向了内容本身(what).

区别于 P2P 系统和 CDN 网络镜像服务器实行网络边缘缓存^[3],CCN 利用节点自身存储功能进行网内缓

存. 数据包沿请求路径原路逐跳返回,且沿途处处缓存(Cache Everything Everywhere, CEE). 这种泛滥式的缓存方式以牺牲节点的缓存空间为代价换取缓存性能的提升,但面对有限缓存容量与庞大数据需求的矛盾,CEE 必然会导致缓存内容更替频繁. 同时,由于数据请求的类 Zipf 分布,即大部分用户请求集中在少数流行对象上^[4],再加上边缘用户规模与网络拓扑结构对请求分布的影响,必然会导致节点工作负载的不对称,引起链路拥塞.

由于缓存流量的多样化和线速执行的要求^[5],节点间有效而简单的协同策略成为研究发展的趋势,如何实现存储资源的优化配给也成为亟待解决的难题.

因此,引入一个有效的负载平衡机制,最大化保留流行内容的多样性和可用性,是发挥 CCN 网内普遍缓存的关键^[6].

2 相关研究

文献[7]通过分析不同的网络拓扑中心度量参数对节点容量做异质化分配,且与同质化网络性能作对比,最终指出异质化的容量分配对全网缓存性能的提升并没有明显作用.文献[8]提出 Betw 策略,数据包沿请求路径返回时只在节点介数最大的重要节点缓存副本.该策略在某种程度上确实缓解了重要节点的缓存替换压力,但也造成了高介数节点过载严重现象.文献[9]改进了 BetwRep 方案,提出结合节点介数和缓存更替率作为综合考量,该方法相比 Betw 获取的性能提升是有限的.文献[10]按细粒度数据块流行度的差异对内容存储器 CS 的空间进行重新分配,缓解了视频流分发过程中产生的链路负载,但通过请求率与缓存占有率的比较实现缓存资源配给,准确性还有待提高.文献[11]提出依据节点介数将任意网络拓扑划分为二层结构,以实现核心节点领导的分块网络之间的协作缓存.但该策略对静态拓扑和核心节点的鲁棒性依赖度高.

以上方案在不同程度上实现了内容的优化存储,但还存在不足之处.实际网络必然需要考虑节点间的联动关系,在全网使用同质化的缓存分配方式,忽视了各个节点独有特征属性.针对上述问题,本文提出一种对热区控制及内容调度的均衡负载方法(Hotspot Control and Content Dispatch, HCCD).先定义了热区节点的评判标准,将拥塞控制问题转化为控制缓存副本放置与保留时间问题.针对同质化缓存考虑因素单一,合理选择执行内容调度的源节点与目的节点,限制需要转移的内容数量,保证重要节点周围存储有尽量多的流行内容,降低缓存替换率的同时提高网内资源的利用率.

3 热区控制及内容调度机制

3.1 合理性证明

本文主要思想是在数据包下行转发过程中,针对负载较大的热区节点,将热门内容调度至相邻相对空闲、负载较低的节点上,尽可能消除热区.因此,首先证明通过内容调度消除热点的同时,可有效换取过载节点缓存性能的提升.

本文的系统模型为一个无向图 $G = (V, E)$, 其中 $V = \{v_1, \dots, v_n\}$ 表示域内的节点集合, $e = (v_i, v_j) \in E$ 为节点 v_i 和节点 v_j 之间的链路.定义 $O = \{o_1, o_2, \dots, o_L\}$ 为完整的对象集,假设对象大小完全相同.每个节点的容量大小 s_i 表示节点的缓存能力.请求到达服从泊松分布 λ

$= \sum_{i=1}^L \lambda_i$, λ_i 是访问 o_i 的内容请求率,定义客户端对内容 $o \in O$ 的请求率为 $\{\lambda_o\}$.

客户端请求时延是描述 CCN 缓存性能的一个重要指标.针对单个节点 V_i 的最佳状况是请求能够被本地缓存的副本响应,这里定义该情况下的接入时延为 t_i .若请求无法在本地响应,但可从 V_i 的邻居节点 V_j 处获取,定义此时时延为 t_j ,表示通过链路从邻端获取内容对象的平均时延.若 V_j 也无法响应请求,则请求被转发至源服务器 S ,由此产生的时延用 t_s 描述.不失一般性,假设 $t_i \leq t_j \leq t_s$.

为便于分析,假设系统只包含两个节点.设节点 V_i 处于过载状态,而 V_j 存储状态为空,并且对象请求率在一定时间内保持稳定.对节点 V_i , t 时刻的缓存命中率 and 请求接入时延分别为:

$$HR_i^t = \frac{\sum_{o \in V_i} \lambda_o}{\sum_{o \in O} \lambda_o} \quad (1)$$

$$latency_i^t = \frac{1}{\sum_{o \in O} \lambda_o} (t_i \sum_{o \in V_i} \lambda_o + t_s \sum_{o \in O \setminus V_i} \lambda_o) \quad (2)$$

为削弱由 V_i 过载导致部分请求无法命中的情况, t_i 时刻 V_i 向 V_j 转移部分内容以分流,令 Δx_{ij} 为超出 V_i 实际负载能力的部分.此时的缓存命中率和请求接入时延为:

$$HR_i^{t_1} = \frac{1}{\sum_{o \in O} \lambda_o} (\sum_{o \in V_i} \lambda_o + \sum_{o \in V_j \setminus V_i} \lambda_o) > HR_i^t \quad (3)$$

$$latency_i^{t_1} = \frac{1}{\sum_{o \in O} \lambda_o} (t_i \sum_{o \in V_i} \lambda_o + \sum_{\Delta x_{ij}; o \in V_j \setminus V_i} \lambda_o t_j + t_s \sum_{o \in O \setminus \{V_i \cup V_j\}} \lambda_o) < latency_i^t \quad (4)$$

由上述公式可得,在消除热区调度内容的过程中,降低请求接入时延的同时系统的缓存命中率也得以提升.令 $k_{jo} \in [0, 1]$ 为每次对 V_j 中调度内容的请求占对总转移对象 Δx_{ij} 请求的比例.若 V_j 为放置 Δx_{ij} 需要腾出部分缓存空间,根据缓存替换算法淘汰对象 $o_r \in V_j$,假设 Δx_{ij} 和 o_r 的请求率分别用 $\lambda_{\Delta x_{ij}}$ 和 λ_{o_r} 表示,且 $\lambda_{\Delta x_{ij}} > \lambda_{o_r}$, 因此 $\Delta x_{ij} \cap o_r = \phi$. 则此时的请求接入时延为:

$$\begin{aligned} w_latency_i^{t_1} &= \frac{1}{\sum_{o \in O} \lambda_o} [t_i \sum_{o \in V_i} \lambda_o + \sum_{o \in V_j \setminus V_i} \lambda_o \sum_{o \in \Delta x_{ij} \setminus \lambda_{o_r}} k_{jo} t_j \\ &\quad + t_s (\sum_{o \in O \setminus \{V_i \cup V_j\}} \lambda_o + \sum_{o \in V_j \setminus V_i} \lambda_o (1 - \sum_{o \in \Delta x_{ij} \setminus \lambda_{o_r}} k_{jo}))] \\ &\quad \sum_{o \in \Delta x_{ij} \setminus \lambda_{o_r}} k_{jo} = 1 \\ &= \frac{1}{\sum_{o \in O} \lambda_o} (t_i \sum_{o \in V_i} \lambda_o + \sum_{\Delta x_{ij}} \lambda_o t_j + t_s \sum_{o \in O \setminus \{V_i \cup V_j\}} \lambda_o) \\ &= latency_i^{t_1} \end{aligned} \quad (5)$$

上式中, $\sum_{o \in V_j \setminus V_i} \lambda_o (1 - \sum_{o \in \Delta x_{ij} \setminus \lambda_{o_r}} k_{jo})$ 表示 V_j 转移出的需向源

服务器 S 请求的对象 o_r . 虽然这部分对象的转移消耗了部分时延,但只要保证替换出的内容具有最小的请求率,就可以把系统消耗的时延控制在最小范围内,甚至平均请求时延反而会大大下降.

以上分析表明,通过内容调度可以削弱高热度节点由于接收请求较多而造成的节点过载、请求拥塞情况. 特别地,在低负载节点创建热门对象的缓存能够更有效地消除热区,防止热区节点由于高替换率使得后期请求无法利用前期缓存.

3.2 热区节点度量

若一个节点位于多个内容分发路径上,意味着该节点拥有较高的兴趣包的来访频率和内容更新频率. 在有限的 CS 资源上加快内容替换频率必然会导致后续请求无法利用前期缓存,而其他节点则可能还有闲置的缓存空间却无法得到利用. 这种负载的不对称性使得网内一部分节点进入过载状态,形成请求“热区”. 因此需要引入负载平衡机制,尽量削弱热区节点与其他节点间的负载差距.

文献[8]中提到中介核心性 (betweenness centrality) 的概念, betweenness 值越大,该节点越容易形成拥塞,变为网络瓶颈. 本文采用节点介数作为衡量节点“热”的度量标准之一,计算公式如下:

$$Bet(i) = \frac{B(i)}{\max_{i,l \in L} \{B(l)\}}, \quad 0 \leq Bet(i) \leq 1 \quad (6)$$

$$B(i) = \sum_{s \neq i \neq t \in V} \frac{\sigma_{s,t}(i)}{\sigma_{s,t}} \quad (7)$$

其中,节点 i 的介数由 $B(i)$ 表示, $\sigma_{s,t}$ 指的是从节点 s 到节点 t 的最短路径数, $\sigma_{s,t}(i)$ 指节点 s 到 t 中经过节点 i 的最短路径数. 由公式 (6) 可得,节点介数 $Bet(i) \in [0,1]$ 是一个相对值,其中 $\max_{i,l \in L} \{B(l)\}$ 指经过节点 i 的请求路径 L 上所有节点介数的最大值.

节点介数依托于网络拓扑结构,表征了节点在网络中的重要程度. 但仅依赖节点介数,无法避免网络参数改变带来的影响. 因此,本文引入另一个判决参数——节点访问度,可以准确描述实时的节点负载情况. 通过对节点重要性和节点自身缓存状况的综合判断,实现对热区节点的最佳筛选.

定义 1 节点访问度 $N_i(t)$. $N_i(t)$ 定义为时间段 T 内节点 i 上到达的实时请求次数,用于表征节点在网络中的活跃程度,体现了用户请求在全网的动态分布情况. 这里采用滑动窗口对请求到达次数进行统计,同时考虑节点的历史请求情况和当前时刻的实时请求数. 图 1 为 $N_i(t)$ 的具体图示.

对节点 i 上请求到达次数进行统计. 图中, T 为滑动窗口宽度, ω 为统计时间间隔. $\gamma_{n-T}, \gamma_{n-T+1}, \dots, \gamma_n$ 分别为时间间隔 $[t_{n-T}, t_{n-T+1}]$, $[t_{n-T+1}, t_{n-T+2}]$, \dots , $[t_n,$

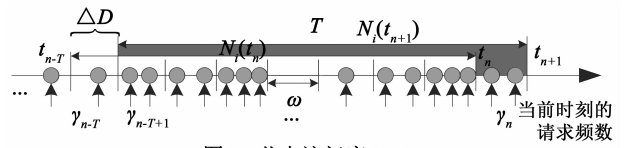


图1 节点访问度 $N_i(t)$

$t_{n+1}]$ 上的请求频数. t_{n+1} 时刻对应的节点访问度 $N_i(t_{n+1})$ 为:

$$N_i(t_{n+1}) = N_i(t_n) + \gamma_n - \Delta D = \sum_{i=n-T+1}^{n-1} \gamma_i + \gamma_n = \sum_{i=n-T+1}^n \gamma_i \quad (8)$$

上式中, $\sum_{i=n-T+1}^{n-1} \gamma_i$ 为历史请求频数和,反映了历史请求对节点访问度 $N_i(t)$ 的影响;而 γ_n 则表征了当前时刻用户的请求分布情况.

定义 2 节点热度 $H(i)$. 基于节点介数和节点访问度这两个判决参量, $H(i)$ 反映了整个网络实时的请求分布情况和负载流量状态,是评价节点是否为热区的重要度量指标.

由于节点介数和节点访问度无法直接关联,归一化衡量标准后:

$$Nor_i(t) = \frac{N_i(t)}{\max_{i,l \in L} \{N_i(t)\}}, \quad 0 \leq Nor_i(t) \leq 1 \quad (9)$$

$H(i)$ 计算公式如下:

$$H(i) = Bet(i) \cdot Nor_i(t) \quad (10)$$

定义 3 节点热阈 δ . 用于表征节点的“热”程度,是判断节点负载状况以及有无溢出的评价指标. 对于网内任意节点 i ,若满足 $H(i) \geq \delta$,即认为该节点为热区节点.

3.3 动态内容调度

根据对节点热度与热阈的定义,数据包上行请求时可快速筛选出热区,以便数据包沿原路径返回时在热区执行内容调度,消除节点间负载不均衡. 由于本文不考虑节点移动性,网络拓扑形成后不再变更,可预先计算出节点介数. 同时,在兴趣包里添加热区标签 $\{R_k, k \in V\}$,以实时记录热点信息.

3.3.1 调度内容选择

由于用户请求具有时间局域性 (temporal locality) 和空间局域性 (spatial locality), LRU (Least recently used) 作为 CCN 中最常用的缓存替换策略,以类似堆栈“先进先出”的模式将“最早”缓存的内容替换出去. 本文以流行度为衡量标准评估对象综合利用价值,避免一致的无差别缓存.

定义统计时间 T 内,对象 c 的流行度抽象为用户对 c 的请求次数占总请求数 N 的比例. 公式如下:

$$f_c(T) = \frac{req_c(T)}{N} \quad (11)$$

文献[12]中提及一种粗略的衰减函数预测方法,如式(12)所示.式中, s 表示由第 n 次增至第 $n+1$ 次请求中,用户请求其他内容的次数, K 为控制衰减的常量.该方法不仅考虑了前次流行度计算,还反映了当前的请求情况.

$$f_{n+1}(c) = f_n(c) \cdot 2^{-\frac{s}{K}} \quad (12)$$

为了选择调度目的节点,排除重复调度与缓存,设计调度信息表(Dispatch Information Table, DIT).当热区执行内容调度后,信息会立即记入 DIT 表,并实时动态更新.具体过程描述如下:

(1) 当节点 v_i 接收到 Data Packet,提取包中的热区标签 $\{R_k, k \in V\}$,若 v_i 在标签内,计算该对象对应的流行度值 $f(c)$.否则,执行 LRU 替换策略.

(2) 将 $f(c)$ 与 v_i 缓存的其他内容的流行度进行比较.若按流行度排序, $Order(f(c))$ 位于前 ε 范围内(这里 ε 取 50%),即认为是热门内容,且 DIT 表中无该内容的调度记录,则执行内容调度.反之,直接在 v_i 上缓存即可.

(3) 提取各邻居的节点访问度 $N_i(t)$,选取具有最小 $N_i(t)$ 值的节点作为内容调度的目的节点.不妨设为 v_j ,即 $v_j = \min \{N_i(t), i \in Nei(v_i)\}$.

(4) 依据 $f(c)$ 计算内容的缓存生命周期 $LT(c)$ (3.3.2 节),并添加至 Data Packet 标签中,随 Data Packet 一起发送给 v_j .同时记录在 DIT 表中.

(5) 邻居 v_j 接收到 Data Packet 后,根据 $LT(c)$ 按正常应答内容进行存储.

3.3.2 调度内容生命周期设置

内容的流行度越大,说明该内容在单位时间内可能被请求的概率就越大.因此,针对内容替换率不高的节点,增加缓存内容的生命周期设置以淘汰部分不流行内容,增大节点的空闲缓存空间.

假定统计时间 T 内,节点上执行内容调度的对象集为 $\{C\}_{i=1}^N$,且各对象对应的流行度为 $\{f(c)\}_{i=1}^N, f(c) \in R^+, \forall i = 1, \dots, N$.设定基础生命周期时长与最长生命周期时长分别为 T_{basic} 与 T_{max} ,则权重系数为 $wc_i =$

$\frac{f_i(c)}{\sum_{j=1, \dots, N} f_j(c)}$.此时,内容的生命周期计算公式如下:

$$LT_i(c) = \begin{cases} wc_i \cdot T_{\text{basic}}, & LT_i(c) < T_{\text{max}} \\ T_{\text{max}}, & LT_i(c) \geq T_{\text{max}} \end{cases} \quad (13)$$

对象的生命期值随数据包一同转发给调度目的节点,参与缓存决策.若发现节点内部有 $LT(c)$ 到期的对象,则删除该内容.

3.3.3 缓存查找与路由转发

HCCD 算法避免了大量的复杂计算和集中化决策.算法 1 和算法 2 分别列出了节点兴趣包和数据包处理的具体执行流程.

兴趣包上行请求数据时,除在 CS 中查询是否已缓

存请求内容外,首先根据 DIT 表判断节点在近期是否已经调度过该内容,避免重复调度.当兴趣包在节点无法响应,且 PIT 表中没有请求内容的对应条目时,查询节点指数 $Bet(i)$,并计算节点访问度 $Nor_i(t)$ 与节点热度 $H(i)$.同时,由热阈 δ 判断节点是否为热区,若是,则将节点标识记入热区标签 $\{R_k, k \in V\}$ 随兴趣包一同转发.

算法 1 Interest Packet 上行发送请求过程

```

For each(  $upstreamv_i \in V$  )
if CS 包含请求内容 then
    返回 Data Packet;
else if DIT 中包含请求内容记录 then
    发送 Interest Packet 到相应邻居调度节点;
else 查询 PIT 表项;
    if 没有请求内容的 PIT 条目 then
        查询  $Bet(i)$ , 计算  $Nor_i(t)$  和  $H(i)$ ;
        if  $H(i) \geq \delta$  then
            在  $\{R_k, k \in V\}$  中记录该节点标识;
            向下一跳节点继续转发 Interest Packet;
        else 已有请求内容的 PIT 条目 then
            在 PIT 中记录到达接口(face)信息;
            提取 Interest Packet 中的热区标签信息;
            丢弃 Interest Packet;

```

算法 2 Data Packet 下行接收响应过程

```

For each(  $downstreamv_i$  )
if  $v_i$  包含在  $\{R_k, k \in V\}$  中 then
    计算 Data Packet 对应流行度  $f(c)$ ;
    if  $Order(f(c)) \geq \varepsilon$ , 且 DIT 表中无调度记录
    then
        选取目的节点  $v_j = \min \{N_i(t), i \in Nei(v_i)\}$ ;
        计算  $LT(c)$ , 随 Data Packet 副本发送给  $v_j$ ;
        在 DIT 表中记录该调度内容;
        继续下行转发 Data Packet 报文;
    else 执行缓存替换策略, 继续下行转发 Data
        Packet;
else 执行缓存替换策略, 继续下行转发 Data Packet;

```

若兴趣包在节点无法响应,但 PIT 表中有请求内容的对应条目.此时在丢弃兴趣包之前,需要将兴趣包的到达接口信息记录到 PIT 表中,并且提取热区标签信息.当相应数据包从该接口转发时,将此标签替换原来数据包携带的热区标签.

数据包按原路径下行转发时,提取热区标签 $\{R_k, k \in V\}$,获取兴趣包上行转发经过的热区节点.由于判断热区条件的计算在请求发送过程中就已完成,数据包在传输过程中无需额外的信息交互.并且针对流行内容请求率高、时效性长的特点,只将流行内容调度至邻居节点.该缓存策略按流行度划分内容,有针对性地平衡热区负载,防止流量向邻居节点过快地无差别调度,失去了 HCCD 算法动态调度的优势.同时,查询 DIT 表中是否有该内容近期已被调度的记录,防止重复调度.

4 仿真与性能分析

4.1 仿真环境与参数设置

本节利用 ndnSIM^[13] 仿真平台对 CCN 网络中不同缓存策略的性能进行对比分析. 网络拓扑采用 GT-ITM 下 Locality 模型生成随机的 50 个路由节点的网络拓扑, 初始节点缓存状态为空, $M=2000$ 个相同大小的内容存储在服务器中, 每个内容划分为 100 个 chunk, 大小为 10KB. 各节点缓存容量一致, CS 设为 100MB, 链路带宽为 100Mbps. 设置 2 个内容源服务器 OCS, 其余节点作为用户接入. 内容请求到达服从 Zipf 分布^[8], 第 i 个内容的请求概率为: $p(i) = C/i^\alpha$, $C = (\sum 1/i^\alpha)^{-1}$. 节点请求到达服从泊松分布, $\lambda = 50$ 个/s. 仿真时间为 300s, 采样周期 $T = 5$ s, 缓存替换策略采用最近最少使用策略 LRU.

4.2 性能分析

将 HCCD 与 CCN(CEE + LRU)^[14] 和 Betw^[8] 进行对比分析. 其中, CCN 作为默认缓存策略是研究 CCN 架构包括命名、缓存、路由与转发, 以及安全隐私的基础, 而 Betw 策略依据节点介数进行缓存, 取得了一定的性能改善. 因此, 这三者的仿真结果有一定的对比价值.

(1) 平均请求时延: 定义为网络中所有节点从发送兴趣包到接收数据包应答的平均时延. 图 2 给出了 $\alpha = 1.2, \lambda = 50$ 时的三种策略平均请求时延的对比情况, 采样周期 $T = 5$ s. 随仿真时间增加, 三种策略的时延均逐渐下降趋于收敛. CCN 由于处处缓存导致各个节点的

缓存替换率居高不下, 因此平均请求时延最大. Betw 只在介数最大的节点缓存内容, 虽然提高了内容的后续利用率, 但也给重要节点带来了极大的负载压力. 同时, Betw 没有考虑节点间的协作缓存, 浪费了大量的存储空间资源, 虽相比 CCN 降低了部分请求时延, 但仍比 HCCD 要高很多. 而 HCCD 在应答过程中, 将流行内容定向存储在路径上最热的节点周围, 实现了就近缓存存储空间资源的有效利用, 因此平均缓存时延最小.

(2) 跳数减少率: 定义为统计时间内实际请求命中时兴趣包经过的总跳数和与在源服务器获得响应的总跳数和之比. 图 3 分别给出了 $\alpha = 1.0, \delta = 0.7$ 和 $\alpha = 1.2, \delta = 0.5$ 时三种策略的跳数减少率对比情况, 采样周期 $T = 5$ s. 随仿真时间增加, 三种策略均呈下降趋势. 初始阶段, 网内各节点刚开始存储内容, 大部分请求还需要转发至源服务器获取, 因此跳数减少率较大. 仿真时间增加后, 请求就近响应的概率加大, 从而跳数减少率也逐渐减小并趋于稳定. CCN 无法针对网络拓扑和节点情况动态缓存, 因此跳数减少率最高. Betw 只考虑拓扑中重要节点, 导致节点间负载差异过大. HCCD 消除热区均衡负载, 通过动态调度削弱了热区节点的替换率, 因此跳数减少率最低.

(3) 缓存命中率体现了所有兴趣包中通过网内缓存获得响应的比例. 网内缓存命中率越高, 兴趣包在靠近请求端获得响应的概率就越大, 减轻源服务器负载的同时, 内容传输跳数也得以减少, 降低带宽消耗.

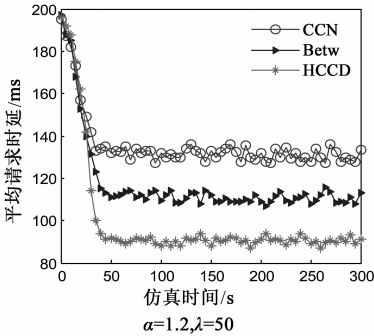


图2 平均请求时延对比

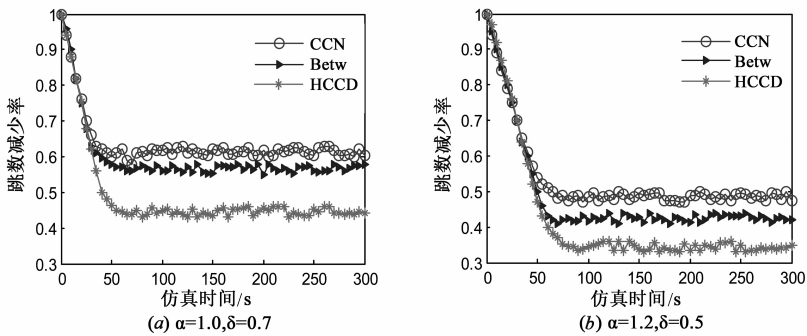
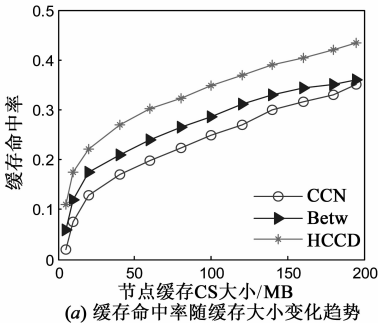
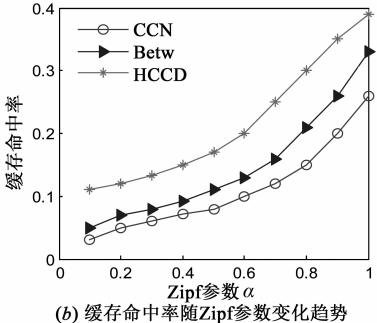


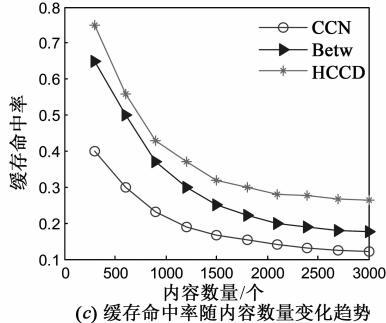
图3 跳数减少率对比



(a) 缓存命中率随缓存大小变化趋势



(b) 缓存命中率随Zipf参数变化趋势



(c) 缓存命中率随内容数量变化趋势

图4 缓存命中率对比

图 4 描述的是在不同情况下缓存命中率变化趋势. 由图 4(a)可以看出, 当节点缓存空间 CS 增大时, 三种策略的缓存命中率都随之增加, 其中 HCCD 增长最快. 由于 HCCD 增大了热门内容的缓存多样性, 从而缓存命中率最高. 图 4(b)反映了缓存命中率随 Zipf 参数的变化情况, Zipf 参数的值越大表示用户对流行内容的请求越集中, CCN 网内缓存特性的优势就越明显. HCCD 的一大目的正是尽可能多地存储流行内容, 因此较其他两种策略有明显优势. 图 4(c)中三种策略的缓存命中率随内容数量的增大都呈现减小趋势. 在节点缓存空间 CS 不变的前提下, 网络总内容数量增多会加快内容在节点的替换频率. HCCD 始终保证热门内容的优先存储, 相比 Betw 和 CCN 受内容数量增多的影响较小.

(4) 缓存分布负载: 为了体现 HCCD 对消除热区、均衡节点负载的效果, 这里对网络中各个节点的负载情况进行考察. 考虑加大缓存策略对负载分布的影响, 将服务器存储总内容数量增加至 8000 个. 图 5 给出了统计时间 200s 内, 请求到达速率 $\lambda = 50$ 个/s 时的各节点累积缓存内容数量. 从图 5 可以看出, 由于 CCN 的处处缓存使得各节点上累积内容数量最多. Betw 只在链路最重要节点缓存内容, 导致个别介数较高的节点上缓存负载要远大于其他节点, 而许多低介数节点几乎没有缓存内容, 造成极大资源浪费. 而 HCCD 通过与邻居节点协作缓存, 消除热区节点的同时增大邻居节点的利用率, 起到了负载均衡的作用.

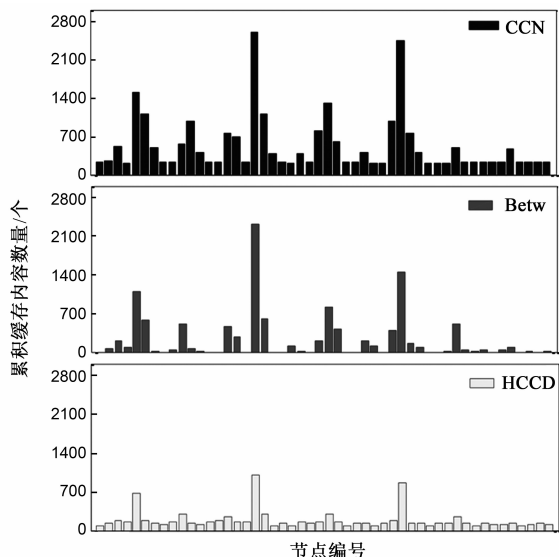


图5 缓存负载分布对比

(5) 适应性讨论: 图 6 给出了 α 三种不同取值情况下, 跳数减少率随热区 δ 变化的趋势图. 当 δ 取值较低时, 即大部分节点都被视为热区节点, 虽然此时的跳数减少率较低, 但过度频繁的调度可能会加重相邻节点

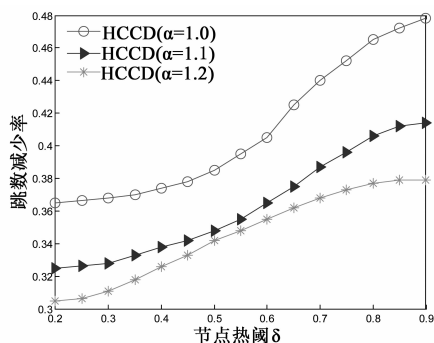


图6 跳数减少率随节点热区 δ 变化趋势

的过载情况, 同时也会引入更多的额外开销, 例如热区标签和 DIT 表的存储开销. 相反的, 当 δ 取值较高时, 能够执行调度的节点范围很小, 限制了 HCCD 算法的优势. 因此, 热区 δ 的确定需要综合权衡缓存性能和开销, 过高或者过低都会违背 HCCD 的设计思想.

5 结束语

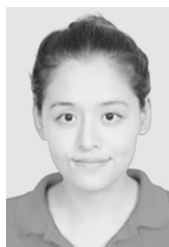
本文针对数据包下行路径中同质化缓存造成的大量冗余, 以及节点资源使用不平衡造成的链路拥塞与热区节点负载过大的问题, 提出了一种热区控制及内容调度缓存算法. HCCD 通过在热区节点自适应地将流行内容调度至邻居空闲节点, 达到了均衡负载和消除热区的作用. 仿真结果表明 HCCD 在减少传输路径和请求时延的同时, 提升了网络资源的总体利用率, 获得了缓存性能的提升. 下一步工作包括: (1) 如何拓展 HCCD 算法, 使之适用中间节点动态变化导致传输失败的情况; (2) 在更复杂的网络规模和仿真参数下进一步验证 HCCD 的缓存性能.

参考文献

- [1] Ahlgren B, Dannewitz C, Imbrenda C, et al. A survey of information-centric networking [J]. IEEE Communication Magazine, 2012, 50(7): 26–36.
- [2] Jacobson V, Smetters D K, et al. Networking named content [A]. Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies [C]. Rome, Italy: ACM, 2009. 1–12.
- [3] 李乔, 何慧, 张宏莉. 内容分发网络研究 [J]. 电子学报, 2013, 41(8): 1560–1568.
Li Q, He H, Zhang H L. Research on content delivery networks [J]. Acta Electronica Sinica, 2013, 41(8): 1560–1568. (in Chinese)
- [4] Breslau L, Cao P, Fan L, et al. Web caching and Zipf-like distributions: Evidence and implications [A]. INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies [C]. New York: IEEE, 1999. 126–134.

- [5] Psaras I, Chai W K, Pavlou G. Probabilistic in-network caching for information-centric networks [A]. Proceedings of the Second Edition of the ICN Workshop on Information-Centric Networking [C]. San Francisco: ACM, 2012. 55-60.
- [6] 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. (in Chinese)
- [7] Dario Rossi, Giuseppe Rossini, et al. On sizing CCN content stores by exploiting topological information [A]. IEEE INFOCOM Workshops on NOMEN [C]. Orlando: IEEE, 2012. 280-285.
- [8] Chai W K, He D, Psaras I, et al. Cache "Less for More" in Information-Centric Networks [M]. Springer Berlin Heidelberg: Networking, 2012. 27-40.
- [9] 崔现东, 刘江, 黄韬等. 基于节点介数和替换率的内容中心网络网内缓存策略 [J]. 电子与信息学报, 2014, 36(1): 1-7.
Cui X D, Liu J, Huang T, et al. A novel in-network caching scheme based on betweenness and replacement rate in content centric networking [J]. Journal of Electronics & Information Technology, 2014, 36(1): 1-7. (in Chinese)
- [10] Li H, Nakazato H, Detti A, et al. Popularity Proportional Cache Size Allocation Policy for Video Delivery on CCN [A]. IEEE European Conference on Networks and Communications (EuCNC) [C]. Paris, : IEEE, 2015. 434-438.
- [11] Xu Y M, Li Y, Lin T, et al. A Dominating-set-based Collaborative Caching with Request Routing in Content Centric Networking [A]. IEEE Conference on Computer Communications [C]. Budapest: IEEE, 2013. 3624-3628.
- [12] Jin S, Bestavros A. Popularity-aware Greedy Dual-size Web Proxy Caching Algorithms [A]. IEEE 20th International Conference on Distributed Computing Systems [C]. Taipei: IEEE, 2000. 254-261.
- [13] Afanasyev A, Moiseenko I, Zhang L. ndnSIM: NDN simulator for NS-3 [R]. NDN, NDN-0005, 2012.
- [14] Kim Y, Yeom I. Performance analysis of in-network caching for content-centric networking [J]. Computer Networks, 2013, 57(13): 2465-2482.

作者简介



方馨蔚 女, 1990 年生于安徽芜湖. 国家数字交换系统工程技术研究中心硕士研究生. 研究方向为宽带信息网、内容中心网络.
E-mail: fxw_1022@sina.com



陈庶樵 男, 1973 年生于黑龙江肇县. 国家数字交换系统工程技术研究中心教授, 硕士生导师. 研究方向为网络体系结构、路由交换技术.
E-mail: chenshuqiao1973@163.com

江逸茗 男, 1984 年生于河南郑州, 国家数字交换系统工程技术研究中心助理研究员, 研究方向为宽带信息网络.

任泽荣 男, 1990 年生于广东佛山, 国家交换系统工程技术研究中心硕士研究生, 研究方向宽带信息网、软件定义网络.