

CCN 中基于替换率的缓存空间动态借调机制

葛国栋¹, 郭云飞², 兰巨龙², 刘彩霞²

(1. 中国人民解放军 75753 部队, 广东 广州 510600; 2. 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(CCN, content centric networking)节点存储资源的有效利用和优化配给问题, 在同质化缓存分配的基础上, 提出了一种基于替换率的缓存空间动态借调机制。该机制从节点存储空间使用状态的动态差异性出发, 首先对于缓存资源借调的合理性给予证明, 进而, 依据节点对于存储资源的需求程度, 动态地执行缓存借调, 将相对空闲的存储资源分配给需求程度更大的节点支配, 换取过载节点缓存性能的提升。该机制减小了内容请求跳数, 提高了缓存命中率, 以少量额外的代价换取了内容请求开销的显著下降, 提升了存储资源整体利用率, 仿真结果验证了其有效性。

关键词: 内容中心网络; 动态借调; 缓存替换率; 异质化分配

中图分类号: TP393

文献标识码: A

Dynamic cache size transfer scheme based on replacement rate in content centric networking

GE Guo-dong¹, GUO Yun-fei², LAN Ju-long², LIU Cai-xia²

(1. Troop 75753, People's Liberation Army, Guangzhou 510600, China;

2. National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China)

Abstract: How to allocate an appropriate cache size to each router and efficiently utilize the finite storage resources poses challenges to the caching performance in CCN (content centric networking). On the basis of homogeneous cache allocation, a dynamic cache size transfer scheme based on replacement rate is proposed. In the scheme, the rationality of the cache size transfer is demonstrated firstly. And then, according to the level of actual demand for the storage resources, the cache transfer procedure is implemented dynamically on the nodes which have different caching states. The idle cache resources are temporarily transferred to the overload node to reduce cache miss rate and enhance the caching gain. The simulation results show that, with a small amount of additional cost, the proposed scheme can efficiently decrease the request hops, increase the cache hit ratio, and improve the overall utilization of storage resources.

Key words: content centric networking; dynamic transfer; cache replacement rate; heterogeneous allocation

1 引言

随着互联网技术与应用的飞速发展,“宽带化”、“内容化”与“个性化”已成为网络发展的主旋律,人们对于数据内容的需求日益强烈,网络应用的主体逐步向内容请求和信息服务演进^[1,2]。据 Cisco VNI Mobile Forecast 预测,到 2014 年互联网

上所有内容相关的流量将占据超过 97.5%的份额,传统的以主机为中心的网络体系结构难以满足当前信息服务的发展要求。为了适应不断增长的数据内容访问需求,信息中心网络(ICN, information-centric networking)^[3]作为一种革命式(clean-slate)的未来互联网设计思路,让数据内容本身成为网络通信的主体单元,将网络通信模式从关

收稿日期: 2014-04-17; 修回日期: 2015-03-13

基金项目: 国家重点基础研究发展计划(“973”计划)基金资助项目(2012CB315901); 国家高技术研究发展计划(“863”计划)基金资助项目(2011AA01A01)

Foundation Items: The National Basic Research Program of China (973 Program)(2012CB315901); The National High Technology Research and Development Program of China (863 Program)(2011AA01A01)

注“在哪”转变为关注“是什么”,即用户和应用通信的目的和意向,成为未来 Internet 设计的重要模式。其中,内容中心网络(CCN, content centric networking)^[4]作为典型的 ICN 结构范例,在中间层用命名数据取代 IP,数据传输采用“发布-请求-响应”模式,直接以内容名字进行路由。节点收到兴趣分组(interest packet)请求后,依据内容名字依次在内容存储器(CS, content store)、未决请求表(PIT, pending interest table)和转发信息库(FIB, forwarding information base)中进行匹配查询,实现点到多点的 content 分发。

在 CCN 的设计中,采用网络内在普遍缓存(in-network caching)的方式,在 interest packet 沿途转发路径(on-path)的所有节点上缓存应答内容,使网络不仅是一个传输体,更是一个内容存储、服务平台。内容的细粒度普遍缓存是 CCN 网络设计的核心特征,通过内容泛在存储,后续请求可以在就近缓存资源处进行应答,有效缩短了用户请求时延,减小了网络数据流量传输。但是,相比海量的内容请求和数据分发,CCN 节点可支配的存储资源,即 CS 的大小十分有限,不可能实现所有应答内容的全部存储。为此,如何充分地利用网络有限的存储资源,实现节点优化配给,是有效发挥 CCN 网络普遍缓存优势的基础和保证^[5]。

目前,针对 CCN 的缓存算法、替换策略、路由转发机制的设计都是在同质化缓存分配(homogeneous cache allocation)的假设基础上研究的^[6,7],即给定网络存储资源,各节点 CS 大小一致,

均匀分配。但是,由于内容请求分布特征、节点位置关系等因素的差异性,不同节点在内容请求中发挥的作用和重要程度不同^[8],节点的缓存空间使用状态,对于存储资源的需求程度也具有较大的差异。如果盲目基于同质化的缓存分配方式,将导致某些重要节点上缓存空间配给不足,引起频繁替换更新,而空闲节点上的缓存资源得不到有效利用,无法实现存储资源优化配给和高效利用。

2 问题分析

同质化的缓存分配方式忽略了节点在内容分发中重要程度的差异性,无法体现节点对于存储资源的需求大小。例如,在图 1 中,节点 *a, b, c, d* 缓存空间 CS 大小相同,均设为 3 个存储单元。由于 *a, b, c* 发送的内容请求如果不能在本节点命中,都需要发送至节点 *d* 进行后续处理和转发,节点 *d* 在内容请求中发挥的作用明显大于其他节点。但是,由于同质化的缓存分配,节点 *d* 仅有的 3 个缓存单元无法满足存储需求,致使内容频繁替换(C_1, C_3)。当后续 interest packet 到达时,已有的缓存内容已被替换淘汰,内容请求只能发送至源服务器(content server)进行响应。此时,如果能依据节点对于存储空间的需求程度,适当增大节点 *d* 的 CS 大小,即使增加 1 个存储单元,也可有效地减小缓存缺失率, C_1, C_3 均可在节点 *d* 的 CS 得到应答。

在传统的 P2P 和 CDN 网络中,缓存资源分配大都建模为多目标的线性规划问题^[9,10],即在给定的层次化网络拓扑结构下,预先计算少数代理节点

-----▶ 兴趣分组请求序列(*d*): $C_1C_3C_2C_4C_1C_3\dots$

■ 缓存替换内容(chunk)

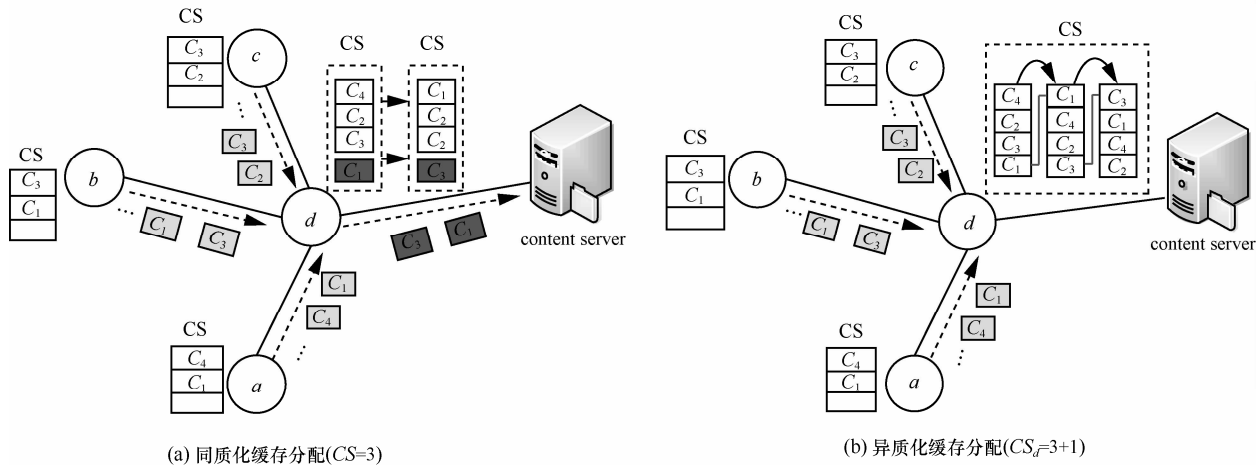


图 1 同质、异质缓存分配对比

或内容服务器的最佳部署位置和存储大小。但是，基于静态的线性规划方法并不适应于任意拓扑关系和节点普遍缓存的 CCN 网络环境；文献[8]在 CCN 中提出了一种基于节点拓扑信息的缓存空间分配策略，并在不同的评价指标(度数中心性、介数中心性等)下进行了分析验证。但是，仅仅依据静态的拓扑位置关系，难以准确地反映节点缓存空间的实时使用状态和对于存储资源的需求大小，而且这种静态的分配方式无法适应节点重要性的动态变化；文献[11]采用数据挖掘的方法，基于多目标学习分析 CCN 网络流量分布特征和用户行为规律，在此基础上将节点划分成不同类别，对应分配不同缓存大小。但是，该方案需要前期大量的数据分析处理。文献[12]提出了基于节点介数的缓存策略(betw)，请求内容只存储在沿途传输路径介数最大的节点上。但是，在同质化缓存分配下，该方案将导致高介数节点出现严重的缓存过载^[13]。文献[14]指出，在 CCN 网络中，由于线速处理速度和存储接入技术的限制，在缓存空间分配时，节点 CS 的大小并不能随意的指定和增加。

针对上述不足，本文认为节点存储空间的大小应依据在内容分发过程中，缓存空间的使用状态和节点对于存储资源的需求程度来决定，并进行动态调节。对于处于频繁替换的缓存过载节点，应提供更多的存储资源。相反，对于相对空闲节点，则需要减小其存储空间大小。为此，针对 CCN 网络，提出了一种缓存空间动态借调机制(DCSTS, dynamic cache size transfer scheme)。DCSTS 在同质化缓存分配的基础上，依据节点缓存替换率，动态地执行缓存资源借调，借助空闲节点的部分存储空间换取过载节点缓存性能的有效提升，增大存储资源的整体利用率。

3 缓存空间动态借调机制

3.1 缓存借调合理性证明

DCSTS 主要思想是在内容请求过程中，依据节点对于存储资源的需求程度，将相对空闲的缓存空间分配给需求程度更大的节点支配，即让高缓存缺失率的过载节点动态地借调相对空闲节点的存储资源，换取缓存缺失率的下降，提升存储资源的整体利用率。为此，首先必须证明，通过节点间缓存空间的借调，高缓存缺失率节点获取的收益大于低缓存缺失率节点性能的损失。

文献[15]给出了 CCN 网络节点缓存缺失概率的在平稳状态下的稳态分布： $\pi_{N+1} = (\frac{\mu}{\lambda + \mu})^N$ 。其中， λ 为内容请求到达率， μ 为缓存内容位置下降率， N 为节点的存储空间。在此，给定节点 v_i 和 v_j ，存储空间 $C(v_i) = C(v_j) = C$ ，令 $m_i = \frac{\mu_i}{\lambda_i + \mu_i}$ ， $m_j = \frac{\mu_j}{\lambda_j + \mu_j}$ 。假设 v_i 为高缓存缺失率节点，即 $\pi_{N+1}(v_i) > \pi_{N+1}(v_j)$ ，则有 $m_i > m_j$ 。 v_i 向 v_j 申请缓存空间借调，大小设为 ΔC 。则问题可描述为：在 $0 < m_j < m_i < 1$ ， $C \in N^+$ 和 $\Delta C = \{0, 1, 2, \dots, C\}$ 的条件下，是否存在 ΔC ，使式(1)成立。

$$m_i^C - m_i^{C+\Delta C} > m_j^{C-\Delta C} - m_j^C \quad (1)$$

其中， $\Delta C = 0$ 表示不执行缓存空间借调； $\Delta C = C$ 表示借用节点全部存储空间。式(1)左侧表示通过缓存空间 ΔC 的借调， v_i 减小的缓存缺失率，即性能提升。右侧表示 v_j 增大的缓存缺失率，即性能损失。

证明 1 缓存借调空间 ΔC 的存在性

为了证明 ΔC 的存在，建立目标函数： $f = (m_i^C - m_i^{C+\Delta C}) - (m_j^{C-\Delta C} - m_j^C)$ ，即要证明在 $0 < m_j < m_i < 1$ ， $C \in N^+$ 和 $\Delta C = \{0, 1, 2, \dots, C\}$ 的条件下，存在 ΔC 使 $f > 0$ 即可。

首先，求解 f 对应的一阶和二阶导函数，有 $f' = -m_i^{C+\Delta C} \ln(m_i + m_j^{C-\Delta C}) \ln m_i$ ， $f'' = -m_i^{C+\Delta C} \ln m_i \ln m_i - m_j^{C-\Delta C} \ln m_j \ln m_j$ 。因为 $0 < m_j < m_i < 1$ ， $\ln m_i$ ， $\ln m_j$ 小于 0， $m_j^{C-\Delta C}$ ， $m_i^{C+\Delta C}$ 大于 0，则对于任意 ΔC ，有 $f'' < 0$ 。所以， f' 为单调递减函数， f 为凸函数。

对于 f' ，由于其单调递减，当 $\Delta C = C$ 时，将取得最小值，即 $f'_{\min} = f'(\Delta C = C) = -m_i^{C+C} \ln m_i + \ln m_j$ 。因为 $-\ln m_i < -\ln m_j$ ， $m_i^{2C} > 0$ ，可得出 $f'_{\min} < 0$ 。当 $\Delta C = 0$ 时， f' 将取得最大值，即 $f'_{\max} = f'(\Delta C = 0) = -m_i^C \ln m_i + m_j^C \ln m_j$ 。假设 $f'_{\max} > 0$ ，可解出：

$$C > \log_{\left(\frac{m_i}{m_j}\right)} \left(\frac{\ln m_j}{\ln m_i}\right)$$

在此条件下， $f'_{\min} < 0$ ， $f'_{\max} > 0$ ，且 f' 为单调递减函数，则 f 将在 ΔC 的取值区间内，先单调递增，然后单调递减，并且为凸函数。

对于 f ，易知，当 $\Delta C = 0$ 时， $f(0) = (m_i^C - m_i^C) - (m_j^C - m_j^C) = 0$ 。所以，在 f 的单调递增区间，

必存在 $\Delta C > 0$, 使 $f(\Delta C) = (m_i^C - m_i^{C+\Delta C}) - (m_j^{C-\Delta C} - m_j^C) > 0$ 。证毕。

证明 2 ΔC 解空间和最优解的存在

由于 f 先单调递增, 再单调递减, 并且 f' 恒小于 0, 那么在 $f'=0$ 对应的 ΔC 处, f 将取得最大值, 获得最大收益。不妨设为 ΔC_{opt} , 即 ΔC 的最优取值。

当 $f'=0$ 时, $f' = -m_i^{C+\Delta C} \ln m_i + m_j^{C-\Delta C} \ln m_j = 0$, 可解出 $\Delta C_{opt} = \log_{(m_i, m_j)} \left(\frac{\ln m_j \left(\frac{m_j}{m_i} \right)^C}{\ln m_i \left(\frac{m_i}{m_j} \right)^C} \right)$ 。

当 $\Delta C = \Delta C_{opt}$, $f(\Delta C_{opt})$ 取得最大值, 接着 f 转入单调递减区间。所以, 当 $\Delta C > \Delta C_{opt}$ 时, 也存在 ΔC 的有效取值, 满足 $f(\Delta C) > 0$ 。但是, 必须注意, 此时 $f > 0$ 的取值均可在 $0 < \Delta C < \Delta C_{opt}$ 范围内, 通过更小的 ΔC 借调而获得。所以, 该部分取值不纳入缓存空间借调 ΔC 取值范围。综上, ΔC 的有效取值范围为 $(0, \Delta C_{opt}]$ 。

综上, 可以获知在 $C > \log_{\left(\frac{m_i}{m_j}\right)} \left(\frac{\ln m_j}{\ln m_i} \right)$,

$\Delta C \in (0, \Delta C_{opt}]$, $\Delta C_{opt} = \log_{(m_i, m_j)} \left(\frac{\ln m_j \left(\frac{m_j}{m_i} \right)^C}{\ln m_i \left(\frac{m_i}{m_j} \right)^C} \right)$ 时,

通过节点间缓存资源的借调, 可以借助空闲节点的部分存储资源来换取过载节点缓存性能的有效提升, 从而增大网络节点缓存资源的整体利用率。

3.2 存储空间需求度量

由于不同节点在内容请求中发挥的作用和重要程度不同, 其 CS 也将表现出不同使用状态: 1) 对于某些重要节点, 内容请求到达率很高, 在有限的存储空间下, 将会引起高频率的缓存替换更新, 最新缓存内容, 可能还没有等到后续内容请求, 已被替换淘汰。后续请求无法有效利用前期缓存, 会进一步增大缓存缺失概率, 形成恶性循环。此时, 对于该类缓存过载节点, 需要提供更多的存储空间; 2) 相反, 对一些相对空闲节点, 内容请求到达率较低, 节点缓存内容得不到有效请求和利用, 浪费了有限的存储资源。对于该类节点, 需要减小存储空间。为此, DCSTS 以缓存替换率来表征 CS 的使用状态, 反映节点对于存储资源的需求程度。

定义 1 缓存替换率 $Rep(v, T)$: 用于表征单位存储资源下, 节点在时间 T 内缓存替换的内容大小。

$$Rep(v, T) = \frac{\sum_{i=1}^{\mu} Chunk(i)}{C(v)} \quad (2)$$

其中, μ 表示时间 T 内, 节点 v 执行的缓存替换次数, $Chunk(i)$ 表示替换的内容大小(单位取 chunk), $C(v)$ 为缓存空间大小。 $Rep(v, T)$ 取值越大, 节点缓存替换频率越高, 对应的请求缺失概率越大。此时, 如果向过载节点能分配更多的存储空间, 则可以有效增大已有缓存驻留概率和后续利用率。 $Rep(v, T)$ 可以有效地表征缓存空间的实时负载和使用状态, 体现了节点对于存储资源的需求程度。

定义 2 缓存借调阈值 δ_B : 用于节点判断是否执行缓存借调过程, 设置为最大容忍缓存替换率。当 $Rep(v) \geq \delta_B$ 时, 节点将执行邻居缓存借调, 申请额外的存储资源来降低缓存替换频率。 δ_B 取值设定涉及具体网络环境、存储空间大小, 内容请求分布等因素。 δ_B 过大将导致只有少数节点才能执行缓存借调, DCSTS 优势不能有效发挥; 而过小的 δ_B 取值又会增大缓存借调引入的额外开销。在仿真实验时, 以 $Rep(v, T)$ 的平均值作为 δ_B 的参考取值, 逐步进行调节, 确定最优 δ_B 取值。

3.3 动态缓存借调

节点记录在时间 T 内, 接收的 Interest packet 数量 λ , 执行的缓存替换次数 μ , 计算对应的 $Rep(v, T)$, 并与 δ_B 进行比较。当 $Rep(v) \geq \delta_B$ 时, 节点将发起缓存借调过程。

3.3.1 借调报文交互

为了选择缓存借调目标节点, 计算对应的借调空间大小 ΔC , 在 CCN 中, 设计了缓存申请(CAP, cache application packet)和缓存响应(CRP, cache response packet)报文, 具体格式如图 2 所示。

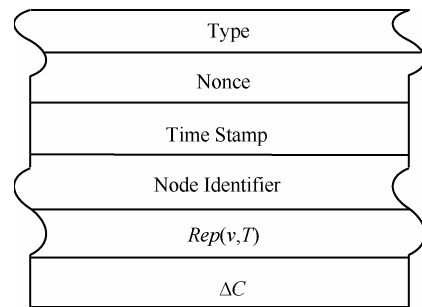


图 2 缓存申请/响应报文

其中, Type 字段表示报文类型, 用于区分 CAP 和 CRP, Nonce 字段为随机数, Time Stamp 字段

用来记录报文发送时间, Node Identifier 为节点标识。

Step1 当 $Rep(v_i, T) \geq \delta_B$, 节点 v_i 向其邻居节点 $Nei(v_i)$ 发送 CAP 报文, 申请缓存空间借调, 其中携带 v_i 对应的 $Rep(v_i, T)$, 由于此时还无法确定目标借调节点和 ΔC 大小, ΔC 字段为 NULL。

Step2 邻居节点接收到 CAP 报文后, 获知 v_i 发起了缓存借调。添加 $Rep(v, T)$ 选项, 发送 CRP 报文进行响应。

Step3 v_i 接收到 CRP 报文后, 提取 $Rep(v, T)$ 取值并进行比较, 选取最小 $Rep(v, T)$ 对应的邻居节点作为缓存借调目标节点, 不妨设为 v_j , 即有 $Rep(v_j, T) = \min\{Rep(v, T), v \in Nei(v_i)\}$ 。

Step4 v_i 依据 $Rep(v_i, T)$, $Rep(v_j, T)$ 的大小关系, 计算借调空间 ΔC (3.3.2 节), 并添加到 CAP 对应的 ΔC 字段, 发送给节点 v_j 。

Step5 v_j 接收到 CAP 后, 依据 ΔC 取值, 将其 CS 划分为借调存储空间(TCS, transferred content store)和归属存储空间(HCS, home content store)2 部分。其中, HCS 归属节点 v_j 支配, 用于正常应答内容的存储, 而对于 v_i 替换淘汰的内容, 将发送至节点 v_j , 在 TCS 中进行缓存。

3.3.2 借调空间 ΔC 计算

节点对应的 $Rep(v, T)$ 越大, 说明其缓存替换频率越高, 需要更大的存储空间来增大已有内容驻留时间和后续利用率。为此, 在 ΔC 确定时, 以 $Rep(v, T)$ 来代表节点对于存储空间的需求程度, 依据 $Rep(v_i, T)$, $Rep(v_j, T)$ 的需求比例, 计算对应 ΔC 大小

$$\begin{aligned} \Delta C &= \left[\frac{(C_i + C_j)}{Rep(v_i, T) + Rep(v_j, T)} \right] Rep(v_i, T) - C_i \\ &= \left(\frac{Rep(v_i, T) - Rep(v_j, T)}{Rep(v_i, T) + Rep(v_j, T)} \right) C \end{aligned} \quad (3)$$

当 ΔC 分配后, 如果 v_i 对应的 $Rep(v_i, T)$ 仍大于阈值 δ_B , 说明借调空间不足。此时, v_i 将继续执行缓存借调过程, 申请更多的存储资源。当后续节点对应的 $Rep(v, T)$ 超出 δ_B 后, 如果该节点的存储空间已被其他节点借用, 则首先减小并释放已借调的 TCS 空间, 减小为: $\Delta C' = \Delta C / 2$ 。若节点存储空间没有被其他节点借用, 则执行正常的缓存借调过程。

3.3.3 缓存查找与路由转发

当 v_i 完成缓存借调后, 后续 CS 替换的内容将会被发送到目标借调节点, 并在其 TCS 中进行存储。同时, 为了维持 TCS 的存储状态, v_i 将建立借调信息表(TIT, transferred information table)来记录 TCS 中存储的内容信息。每当替换内容被发送到 TCS 存储后, 对应的内容名字将被添加到 TIT 表项中, 并进行动态更新。当 v_i 后续接收到 interest packet 后, 首先查询其缓存空间 CS 是否含有该请求内容, 如果命中, 直接发送 data packet 进行响应。如果没有, 则查询 TIT 表项, 若包含该请求内容名字, 则说明目标借调节点的 TCS 中存储了该请求内容, 将 interest packet 转发到目标借调节点, 由 TCS 存储的内容进行应答。否则, 按照 FIB 表项执行下一跳路由转发。算法 1 给出了 DCSTS 的整体流程。

算法 1 基于替换率的缓存空间动态借调机制 (DCSTS)

初始化: $G(V, E)$, δ_B , $Rep(v_i, T)$, T

缓存借调: CAP, CRP

1) **foreach**($v_i \in V = \{v_1, v_2, \dots, v_N\}$)

2) 周期计算 $Rep(v_i, T)$;

3) **if** $Rep(v_i, T) \geq \delta_B$

4) **if** v_i 的 CS 已被借调, TCS = ΔC

then

5) 释放借调空间 TCS: $\Delta C' = \Delta C / 2$;

6) **else**

7) **While**($Rep(v_i, T) \geq \delta_B$)**do**

8) 选择目标借调节点 v_b ,

$Rep(v_b, T) = \min\{Rep(v, T)\}$;

9) 计算缓存借调空间 ΔC ;

10) 划分存储空间 CS: TCS = ΔC ,

HCS = $1 - \Delta C$;

11) **end while**

内容请求与应答: interest packet, data packet

12) **if** v_i 缓存空间已满 **then**

13) 执行缓存替换, 删除最底层缓存单元内容 C_i ;

14) 将 C_i 发送到目标借调节点 v_b 的 TCS 中进行存储;

15) **foreach**(v_i 接收到的 interest packet)

16) **if** CS 包含请求内容 **then**

17) 发送 data packet 直接进行响应;

18) **else** 查询 TIT 表项

19) **if** TIT 包含请求内容名字 **then**
 20) 转发 interest packet 到目标借调
 节点 v_b ;
 21) **else** 按照 FIB 表项执行 interest
 packet 下一跳路由转发

4 仿真与性能分析

4.1 仿真环境与参数设置

采用 ndnSIM^[16, 17]进行仿真与性能分析, 该工具对于 CCN 的基本数据单元结构和路由转发流程均已实现, 并提供了开放的源码和运行实例。在 GT-ITM 下采用 Locality 模型生成 50 个路由节点的平面随机网络拓扑, 图 3 给出了仿真拓扑在 NS-3 python 下的可视化显示。网络中内容对象总数 N 为 2 000 个, 内容序号以 1~2 000 依次排序, 每个内容划分为 100 个 chunk, 大小设为 10 kB。节点缓存容量一致, CS 设为 200 MB, 可存储 200 个请求内容, 链路带宽 100 Mbit/s。在网络中设置 2 个内容服务器, 负责内容对象的存储和发布, 各服务器随机存储 1 000 个内容, 并在网络边缘随机选取 2 个节点与内容服务器直接相连。其余节点作为用户接入节点, 发布内容请求, 内容请求到达服从参数 $\lambda=20$ 个/s 泊松过程^[12], 请求概率服从 Zipf 分布^[12]。为了构造内容请求的动态变化和的局域分布特征, 在内容请求对应 Zipf 分布序列中随机选取一定比例的扰动内容, 进行随机化重新排序^[18], 扰动比例选取 5%, 即 100 个内容对象。仿真时间为 300 s, 采样周期 $T=5$ s。初始节点缓存状态为空, 无任何

内容副本存储, 缓存替换策略为最近最少使用策略 LRU。

4.2 性能分析

将 DCSTS 与同质化缓存分配的 CCN(Ho-CCN)^[6]、基于度数中心性(DC, degree centrality)的异质缓存分配方案(He-DC)^[8]进行对比分析, 性能评价指标包括: 路径减小率(PRR, path reduction ratio), 缓存命中率(CHR, cache hit ratio)和额外开销对比。

4.2.1 路径减小率

路径减小率 PRR : 兴趣分组实际命中经过的路由跳数(hop)与在源服务器端获取响应传输的路由跳数(Hop)之比

$$PRR = \frac{\sum_{i=1}^{Interest_NUM(T)} hop_i}{\sum_{i=1}^{Interest_NUM(T)} Hop_i} \quad (4)$$

其中, $Interest_Num(T)$ 表示时间 T 内, 发送的兴趣分组数量。当兴趣分组在首跳接入节点命中时, $hop=0$, $PRR=0$ 。当兴趣分组转发至源服务器进行响应时, $hop=Hop$, $PRR=1$ 。

图 4 分别给出了在: 1) $\alpha=0.8$, $\delta_b=0.70$; 2) $\alpha=1.0$, $\delta_b=0.55$ 时, 3 种方案的 PRR 对比, 采样间隔 $T=5$ s。仿真初始阶段, 由于节点存储状态为空, 发送的兴趣分组请求都需要转发至源服务器进行响应, PRR 较大。但随着内容的不断存储, 缓存内容的响应概率逐渐增加, PRR 随之减小。

对于 Ho-CCN, 同质化的缓存分配方式缺乏对于节点重要性和缓存使用状态的考虑, 致使缓存过载节点出现频繁的替换更新, 后续请求无法利用前

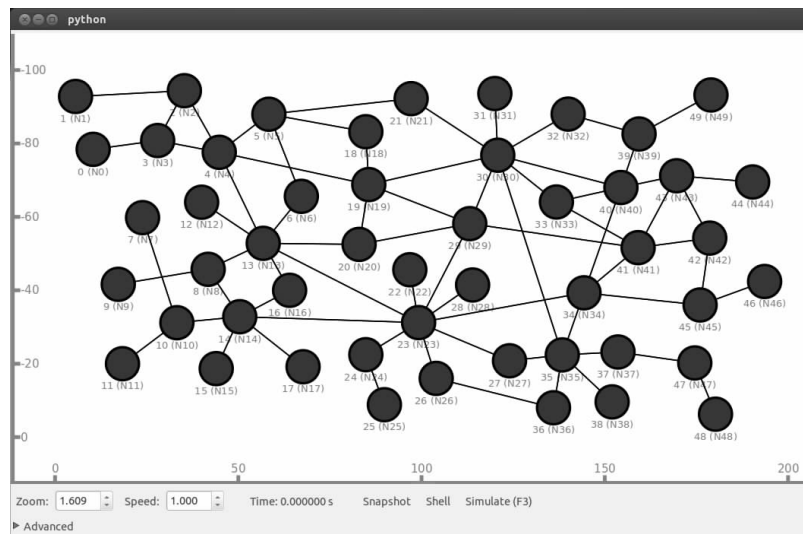


图 3 网络仿真拓扑

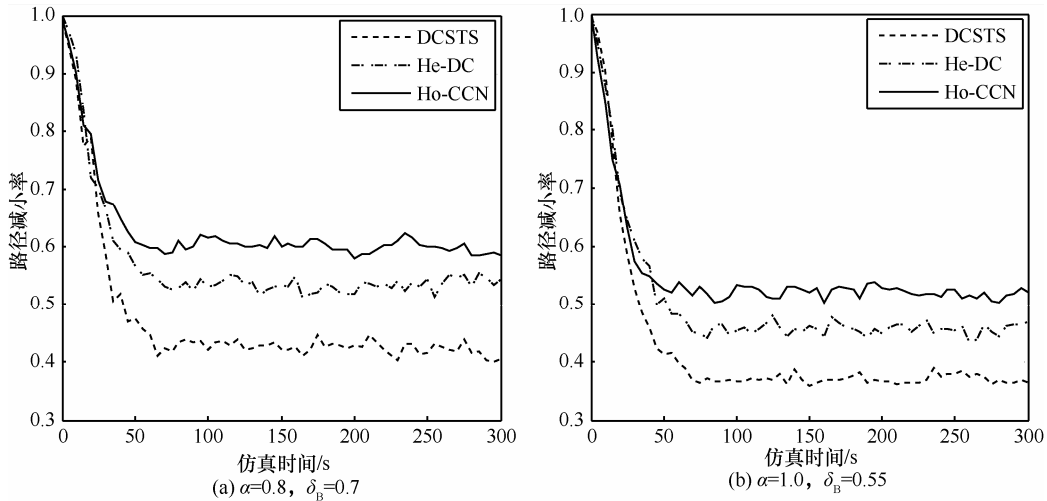


图4 路径减小率 PRR 对比

期缓存, 增大了缓存缺失概率和兴趣分组传输路径, PRR 最大; 在 Ho-DC 中, 依据 DC 来判断节点重要程度并执行差异化的存储空间分配。但是, 仅仅依据节点的拓扑位置关系, 并不能实时准确地反映节点缓存空间的使用状态, 加之内容请求分布的动态变化, 这种预先的静态分配策略无法实现后续动态调节, 性能提升有限($\alpha=0.8$: 8.9%, $\alpha=1.0$: 7.6%); DCSTS 依据节点实时的 $Rep(v, T)$, 来判断节点对于存储资源的需求程度, 动态借调空闲节点的存储资源来减小缓存过载节点的替换率, 增大已有内容驻留概率和利用率, 有效减小了内容传输跳数($\alpha=0.8$: 28.3%, $\alpha=1.0$: 25.4%), PRR 是各方案中最小的。

4.2.2 缓存命中率

缓存命中率 CHR : 网络中节点缓存内容响应兴趣分组请求的概率。缓存内容命中率越高, 源服务器响应率和负载就越小。定义缓存命中函数 δ_i 为

$$\delta_i = \begin{cases} 1, & h_i = \text{cache} \\ 0, & h_i = \text{server} \end{cases} \quad (5)$$

其中, δ_i 为 1 表示 interest packet 在节点缓存处得到应答, δ_i 为 0 代表内容请求在内容服务器处进行响应。在时间 T 内, 缓存命中率 CHR 为

$$CHR = \frac{\sum_{i=1}^{Interest_NUM(T)} \delta_i}{Interest_NUM(T)} \quad (6)$$

图 5 给出了在: 1) $\alpha=0.8$, $\delta_b=0.70$; 2) $\alpha=1.0$, $\delta_b=0.55$ 时, $T=5$ s, 整个仿真时间内各节点的 CHR 对比。

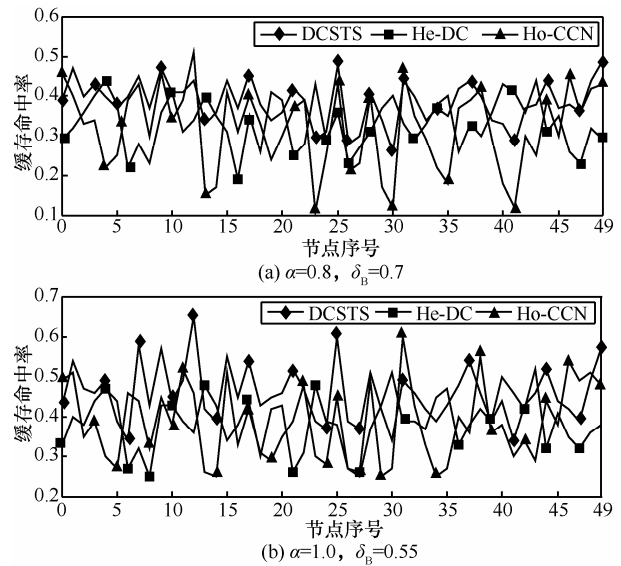


图5 节点缓存命中率对比

对于 Ho-CCN, 同质化的缓存分配方式无法实现存储资源的合理划分, 致使各节点 CHR 出现较大的差距和波动($\alpha=0.8$: 0.11~0.51, $\alpha=1.0$: 0.25~0.60)。缓存过载节点存储空间配给不足, 引发频繁替换, 而空闲节点的缓存空间又得不到有效请求和利用; 在 He-DC 中, 预先基于 DC 进行缓存空间分配, 当节点的重要程度改变后, 无法进行动态调节。特别是当度数较小的一般性节点(regular node)内容请求速率增大, 在实际内容请求中重要性加强时, 过小的缓存空间将会导致严重的缓存缺失; 相比 He-DC, DCSTS 通过动态的执行缓存借调, 将相对空闲的存储空间分配给需求程度更大的节点支配, 以牺牲空闲节点小额的缓存命中率来有效减

小缓存过载节点的内容替换率,降低了各节点 CHR 的差异,提升了网络存储资源整体利用率。图6给出了节点平均 CHR 的对比。

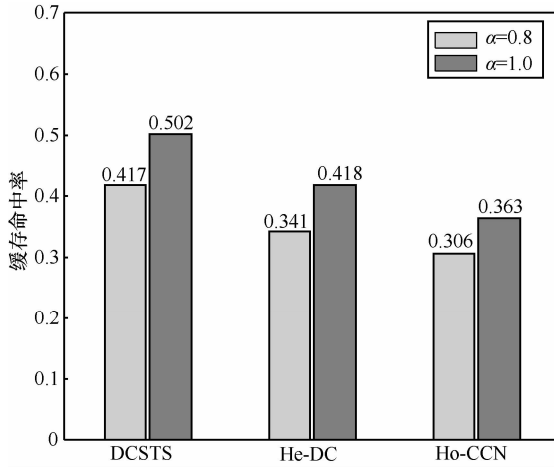


图6 平均缓存命中率对比

4.2.3 代价开销对比

相比 Ho-CCN, He-DC, DCSTS 为了实现缓存空间的动态借调,引入了额外的代价开销,主要包括缓存借调开销(C_B),节点存储开销(C_C)。

1)缓存借调开销(C_B):节点在执行缓存借调时,发送 CAP 和 CRP 报文交互引入的额外代价开销,定义为报文长度与其传输距离(以路由跳数衡量)的乘积,代价单位取 bit·hop。单位时间 T 内缓存借调开销 C_B 为

$$C_B = \sum_{v \in V_B} (P_{CAP}d_1 + P_{CRP}d_2) \quad (7)$$

$$V_B = \{v | v \in V \cap Rep(v, T) \geq \delta_B\}$$

其中, V_B 为执行缓存借调的节点集合, P_{CAP} 、 P_{CRP} 分别表示 CAP 和 CRP 报文长度, d_1 、 d_2 为对应传输的路由跳数。

2) 节点存储开销(C_C): 在 DCSTS 中,节点完成缓存空间借调后,需要建立 TIT 表项,来记录 TCS 的内容存储状态。当节点将缓存替换内容发送至借调节点 TCS 进行存储时,同时将内容名字添加到 TIT 表项中。 C_C 大小取决于 TIT 存储的内容名字长度和数量,代价单位为 bit。

$$C_C = \sum_{i=1}^m l_{name} m \quad (8)$$

其中, l_{name} 表示内容名字长度, m 为对应的存储数量。

3) 内容请求开销(C_R): 定义为内容请求过程

中, interest packet 和 data packet 分别与其传输距离的乘积之和,大小取决于内容请求传输的路由跳数,代价单位取 bit·hop。

$$C_R = (P_{Int} + P_{Dat})d \quad (9)$$

其中, P_{Int} , P_{Dat} 分别表示 interest packet 和 data packet 报文长度, d 为对应的路由传输跳数。

表1给出了 $\alpha=0.8$ 时, $T=5$ s, 各方案的平均代价开销对比。在 Ho-CCN 中,同质化的缓存分配方式不会引入额外的 C_B 和 C_C 。但是在内容请求时,由于无法实现存储资源的优化配给,对应的内容请求开销 C_R 最大。对于 He-DC,依据度数中心性,预先已计算出各节点对应的缓存空间大小,无需动态调整, C_B 和 C_C 为 0; DCSTS 依据节点的实时缓存替换率,动态地执行缓存借调,虽然引入了额外的 C_B 和 C_C ,但有效地减小了内容请求开销 C_R 。而且,缓存借调只涉及局域的一跳邻居节点交互,有效限制了 C_B 的大小。同时,相比存储内容本身, TIT 存储内容名字引入的额外存储开销 C_C 降低了 3 个数量级。DCSTS 通过少量额外 C_B , C_C 的付出,减小了节点的缓存替换率,换取了 C_R 的大幅下降。表2给出在 $\alpha=1.0$ 时,各方案代价开销对比。随着 α 指数的增大,请求内容更加集中在少数流行资源上,各方案的内容请求开销 C_R 都得到不同程度减小。

表1 代价开销对比($\alpha=0.8$)

开销	方案			
	Ho-CCN	He-DC	DCSTS	
			$\delta_B=0.6$	$\delta_B=0.7$
C_B	0	0	621×10^4	403×10^4
C_C	0	0	452×10^3	271×10^3
C_R	5.769×10^8	4.985×10^8	3.965×10^8	4.134×10^8

表2 代价开销对比($\alpha=1.0$)

开销	方案			
	Ho-CCN	He-DC	DCSTS	
			$\delta_B=0.45$	$\delta_B=0.55$
C_B	0	0	510×10^4	319×10^4
C_C	0	0	301×10^3	194×10^3
C_R	4.715×10^8	4.194×10^8	3.228×10^8	3.357×10^8

4.3 适应性讨论

图7给出了在 α 取值为 0.8、1.0、1.2 时, PRR

随借调阈值 δ_b 取值的变化趋势。随着 δ_b 的增大, 执行缓存借调的阈值越高, 借调节点之间缓存替换频率差值越大, ΔC 所带来的额外收益越高, 但是执行缓存借调的节点数量会越少。一些节点虽然已处于较高的缓存替换状态, 但是由于 δ_b 取值过大, 无法满足借调条件, 限制了 DCSTS 优势的有效发挥。相反, 当 δ_b 取值较小时, 将会触发更多的节点执行缓存借调过程, 虽然减小了 PRR , 但同时会引入更多的额外借调开销 C_b 和存储开销 C_c 。而且, 随着 δ_b 取值减小, 借调节点之间缓存替换率差值降低, 相同借调空间 ΔC 所换取的收益不断减小。特别是当 δ_b 取值过小时, 将会导致几乎所有节点执行缓存借调过程, 额外开销急剧增大。此时, ΔC 所带来的性能提升十分有限, 这也违背 DCSTS 缓存过载节点借调空闲节点存储资源的设计思想。所以, 在 δ_b 确定时, 需要同时考虑缓存借调所带来的性能提升和额外引入的代价开销。

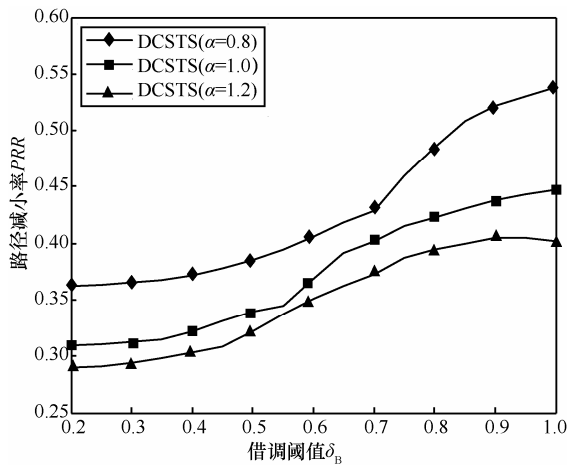


图7 路径减小率 PRR 随借调阈值 δ_b 的变化趋势

图8给出了 PRR 随 Zipf 指数 α 的变化趋势。当 α 取值较小时, 内容请求分布不能有效集中, 在 α 取值为 0.2 和 0.4 时, 最大流行内容的请求概率仅为 0.001 8 和 0.006 3, 多样化的内容请求和存储将导致有限存储空间高频率的替换更新, 增大了 PRR 。随着 α 参数的增大, 内容请求的集中性和局域性不断加强, 流行资源在缓存中的驻留概率和响应率明显增大, PRR 逐步减小。特别是当 α 取值位于 0.8 到 1.2 区间时, 内容请求的集中性急剧增大, 累加概率为 0.8 包含的内容对象数量从 831 项减小到 99 项, PRR 显著下降。但是, 随着 α 取值的进一步增大 ($\alpha > 1.2$), 流行资源对应的内容分布已无

显著变化, PRR 下降趋势逐渐减缓。DCSTS 通过动态的缓存空间借调, 有效利用了节点有限的存储空间, 对于 α 取值变化具有良好的适应性。

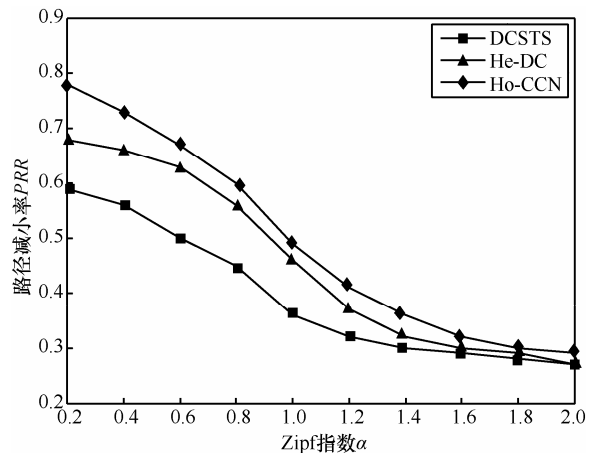


图8 路径减小率 PRR 随 Zipf 指数 α 的变化趋势

5 结束语

为了实现节点存储资源的优化配给和有效利用, 结合缓存空间使用状态的差异性, 提出了基于缓存替换率的节点存储空间动态借调机制。DCSTS 通过节点间动态的缓存资源借调, 将相对空闲的存储资源分配给需求程度更大的节点支配, 换取过载节点缓存性能的有效提升。DCSTS 借助少量额外代价的付出, 提升了网络存储资源的整体利用率, 仿真结果和对比分析显示了其有效性。后续研究工作包括: 1) 如何在 DCSTS 基础上, 设计 CCN 相应的缓存算法、替换策略等; 2) 在不同网络和仿真参数条件下对于 DCSTS 性能和开销进行进一步分析验证。

参考文献:

- [1] XYLOMENOS G, VERVERIDIS C, SIRIS V, *et al.* A survey of information-centric networking research[J]. IEEE Communications Surveys & Tutorials, 2014, 16(2): 1024-1049.
- [2] 兰巨龙, 程东年, 胡宇翔. 可重构信息通信基础网络体系研究[J]. 通信学报, 2014, 35(1): 128-139.
LAN J L, CHENG D N, HU Y X. Research on reconfigurable information communication basal network architecture[J]. Journal on Communications, 2014, 35(1): 128-139.
- [3] AHLGREN B, DANNEWITZ C, IMBRENDA C, *et al.* A survey of information-centric networking[J]. IEEE Communication Magazine, 2012, 50(7): 26-36.
- [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] 朱轶, 糜正琨, 王文鼎. 一种基于内容流行度的内容中心网络缓存概率置换策略[J]. 电子与信息学报, 2013, 35(6): 1305-1310.
ZHU Y, MI Z K, WANG W N. A cache probability replacement policy based on content popularity in content centric networks[J]. Journal of Electronics & Information Technology, 2013, 35(6): 1305-1310.
- [6] KIM Y, YEOM I. Performance analysis of in-network caching for content-centric networking[J]. Computer Networks, 2013, 57(13): 2465-2482.
- [7] ROSSINI G, ROSSI D. A dive into the caching performance of content centric networking[A]. IEEE 17th International Workshop on CAMAD[C]. Barcelona, Spain, 2012. 105-109.
- [8] ROSSI D, ROSSINI G. On sizing CCN content stores by exploiting topological information[A]. IEEE INFOCOM workshop on NOMON[C]. Orlando, USA, 2012. 280-285.
- [9] LAOUTARIS N, ZISSIMOPOULOS V, STAVRAKAKIS I. On the optimization of storage capacity allocation for content distribution[J]. Computer Networks, 2005, 47(3): 409-428.
- [10] KAMIYAMA N, KAWAHARA R, MORI T, *et al.* Optimally designing capacity and location of caches to reduce P2P traffic[A]. IEEE International Conference on Communication[C]. Cape Town, South Africa, 2010. 1-6.
- [11] XU Y M, LI Y, LIN T, *et al.* A novel cache size optimization scheme based on manifold learning in content centric networking[J]. Journal of Network and Computer Applications, 2014, 37: 273-281.
- [12] CHAI W K, HE D, PSARAS I, *et al.* Cache “less for more” in information-centric networks[A]. Proceedings of IFIP Networking[C]. Prague, Czech, 2012. 27-40.
- [13] 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.
- [14] PERINO D, VARVELLO M. A reality check for content centric networking[A]. ACM SIGCOMM Workshop on Information-Centric Networking[C]. Toronto, Canada, 2011. 44-49.
- [15] PSARAS I, CLEGG R G, LANDA R, *et al.* Modelling and evaluation of CCN-caching trees[A]. Proceedings of IFIP Networking[C]. Valencia, Spain, 2011. 78-91.
- [16] AFANASYEV A, MOISEENKO I, ZHANG L X. ndnSIM: NDN simulator for NS-3[R]. NDN, Technical Report NDN-0005, 2012.
- [17] NS-3 based named data networking (NDN) simulator[BE/OL]. <http://ndnsim.net>.
- [18] GUO S, XIE H Y, SHI G. Collaborative forwarding and caching in content centric networks[A]. Proceedings of IFIP Networking[C]. Prague, Czech Republic, 2012. 41-55.

作者简介:



葛国栋 (1985-), 男, 陕西咸阳人, 中国人民解放军 75753 部队工程师, 国家数字交换系统工程技术研究中心博士生, 主要研究方向为网络路由与安全技术。

郭云飞 (1963-), 男, 河南郑州人, 国家数字交换系统工程技术研究中心教授、博士生导师, 主要研究方向为新型网络体系结构、移动互联网。

兰巨龙 (1962-), 男, 河北张家口人, 国家数字交换系统工程技术研究中心教授、博士生导师, 主要研究方向为可重构新型网络体系结构、高性能网络路由和交换技术。

刘彩霞 (1974-), 女, 山东烟台人, 国家数字交换系统工程技术研究中心副教授、硕士生导师, 主要研究方向为新型网络体系结构、移动互联网。