单位代码：10293 密 级：_____

南京邮电大学

专 业 学 位 硕 士 论 文

论文题目：未来网络存储方案性能分析与仿真

学 号 1213012419

姓 名 钱 友

导 师 杨龙祥 教授

专业学位类别 工程硕士

类 型 全 日 制

专业（领域） 电子与通信工程

论文提交日期 二零一六年二月

Performance Analysis and Simulation on Cache Mechanism of the Future Network

Thesis Submitted to Nanjing University of Posts and Telecommunications for the Degree of Master of Engineering

By

You Qian

Supervisor: Prof. Longxiang Yang

February 2016

南京邮电大学学位论文原创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得南京邮电大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

本人学位论文及涉及相关资料若有不实，愿意承担一切相关的法律责任。

研究生签名：_____ 日期：_____

南京邮电大学学位论文使用授权声明

本人授权南京邮电大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档；允许论文被查阅和借阅；可以将学位论文的全部或部分内容编入有关数据库进行检索；可以采用影印、缩印或扫描等复制手段保存、汇编本学位论文。本文电子文档的内容和纸质论文的内容相一致。论文的公布（包括刊登）授权南京邮电大学研究生院办理。

涉密学位论文在解密后适用本授权书。

研究生签名：_____ 导师签名：_____ 日期：_____

摘要

随着互联网技术的快速发展，以及互联网用户数量的日益增加，传统互联网端到端的通信模式越来越难以满足用户的需求，尤其是海量内容的可靠传输和高质量服务保证。为了从根本上解决上述难题，近年来研究者们提出了一系列未来网络体系架构，这些体系架构中的一个普遍而重要的特征就是使用缓存机制，缓存机制不仅能够提高用户请求的命中率、减少用户请求的时延，还能降低网络流量，缓存机制研究的重要性日益显著。

本文分析了两种未来网络体系架构—信息中心网络和延迟容忍网络下的缓存机制。信息中心网络（ICN）中，本文重点研究了一种数据块缓存位置与搜索方案，通过在路由节点中添加一个缓存记录，用于记录节点缓存内容副本情况，保证传输路径上同样的缓存内容只有一份，同时优化路径外请求搜索，根据当前节点到内容服务器的跳数和所设置的阈值的比较，决定将请求向上游还是向下游进行转发。最后，通过仿真实验，在缓存命中率、缓存命中距离、平均下载时延和带宽消耗这些性能上，将该方案与常用缓存放置策略进行对比，证明了该方案的高效性。

延迟容忍网络（DTN）中，由于网络的间歇性连接特点，DTN 采用“存储—携带—转发”的方式传输数据，路由算法是 DTN 研究的关键所在，而缓存能够极大地影响路由算法。本文分析了一种基于消息传播状态的节点缓存管理方案，该方案能够抑制网络中消息副本较多且传播速度较快的消息的进一步扩散，给副本数量较少、传播速度较慢的消息提供了更多被传输的机会。仿真结果表明，相较于其他常见缓存替换和调度策略，基于消息传播状态的缓存管理方案能够提高消息的交付比率，且开销比率和平均时延均优于常见缓存替换和调度策略。

关键词：未来网络，信息中心网络，延迟容忍网络，缓存存储方案

Abstract

With the rapid development of Internet technology, as well as the increasing number of Internet users，the traditional end-to-end communication mode is becoming more and more difficult to meet the needs of users, particularly reliable transmission of large quantities of content and high quality services. In order to solve the fundamental problem, recent years, researchers have proposed a series of future network architectures. One common and important feature of these architectures is to use caching mechanism, Caching mechanism can not only improve the hit rate and reduce the delay of user request, but also reduce network traffic, the study of cache mechanism is becoming increasingly important.

In this thesis, the two implementations of future network-information-centric network and delay-tolerant network caching mechanisms are analyzed. In Information-Centric Network(ICN), this thesis focuses on a chunk caching location and search scheme(CLS), by adding a cache trail which is used to record the data copy of the content in the router, to ensure that there is only one copy of the content on transmission path, while optimizing off-path searching. Comparing the number of hops between current node and the content server to a threshold set before, which used to decide weather forward the request to upstream or downstream. Finally, the simulation experiments show that, CLS is better than conventional caching strategies on caching hit ratio, caching hit distance, average download latency and bandwidth consumption.

In Delay-Tolerant Network(DTN), due to intermittent connection, DTN use “storage-carry-forward” to transfer data, routing algorithm is the key to DTN research, while caching can greatly affect the routing algorithm. This thesis analyzes a caching management based on message transmission status, the scheme can suppress the further spread of the message which has more copies and spreads faster, giving more opportunity to the message which has less copies and spreads slower in the network. The simulation results show that the caching management is better than other commonly used caching strategies on delivery ratio, overhead ratio and average delay.

Key words:future network, information-centric network, delay-tolerant network, caching storage scheme

目录

第一章 绪论.....1
1.1 研究背景.....1
1.2 研究现状.....2
1.3 主要工作和论文结构.....6
第二章 未来网络缓存方案比较研究.....8
2.1 信息中心网络体系架构.....8
2.1.1 命名数据网络体系架构.....8
2.1.2 面向数据传输的网络架构体系.....11
2.2 延迟容忍网络架构.....13
2.2.1 延迟容忍网络的特点.....13
2.2.2 延迟容忍网络体系结构.....13
2.3 信息中心网络中常见缓存策略.....15
2.4 DTN 中常用的缓存策略.....20
2.5 本章小结.....22
第三章 信息中心网络下数据缓存位置与搜索方案.....23
3.1 引言.....23
3.2 数据块缓存位置与搜索策略.....24
3.2.1 CLS 基本操作步骤.....24
3.2.2 CLS 方案中的特殊场景.....26
3.3 对比策略.....27
3.4 仿真实验.....29
3.4.1 仿真工具.....30
3.4.2 仿真环境.....32
3.4.3 仿真结果.....33
3.5 本章小结.....36
第四章 DTN 下基于缓存消息传播状态的存储方案.....37
4.1 引言.....37
4.2 系统模型与分析.....37
4.2.1 系统模型.....37
4.2.2 系统分析.....38
4.3 缓存替换与调度方案.....40
4.3.1 缓存替换方案.....40
4.3.2 缓存调度算法.....41
4.4 仿真实验.....43
4.4.1 仿真环境设置.....43
4.4.2 仿真结果.....43
4.5 本章小结.....49
第五章 总结与展望.....50
参考文献.....52
致谢.....55

第一章 绪论

1.1 研究背景

随着互联网技术与应用的快速发展，互联网流量呈爆炸式增长。据 Cisco 公司 VNI 评估[1]，过去五年，全球 IP 流量增长了 5 倍，并且在接下来的 5 年将增长近 3 倍。2016 年以后全球 IP 流量将超过 1ZB（1000EB）/年，到 2019 年将达到 2ZB。2014 年与内容相关的网络视频流量占用户网络流量的 64%，而到 2019 年将达到 80%。

然而，设计互联网的最初主要目的是为了解决两个主机之间的点到点通信问题，传统互联网这种端到端的通信模式越来越难以满足人们的需求，在可扩展性、移动性和安全性等方面存在着问题：

（1）可扩展性问题。IPv4(Internet Protocol version 4，IPv4)地址面临枯竭的问题，尽管理论上它有 43 亿多个地址，但是世界范围内互联网地址分配的不公平性和不合理的规划，导致全球互联网新的路由表飞速增长。虽然互联网工程任务组（Internet Engineering Task Force，IETF）开发的 IPv6 具有海量的地址空间，但由于缺乏高效路由寻址和安全问题，一直未大规模商用。

（2）移动性问题。互联网的命名是基于主机的 IP 地址的，为了解决路由的可扩展性问题，采用地址分层的结构，命名结构与位置相关联。然而，网络节点的动态性特点越发增强，使得数据传输路径频繁变更，这些严重破坏了上层应用服务的连续性，影响了用户的互联网服务质量体验。

（3）安全性问题。当今互联网是通过在协议上添加安全方式来保障安全的，互联网建立之初，用户节点少并认为这些节点是安全可信的。但随着互联网的爆炸式增长，用户数量急剧增加，网络结构日益复杂，现有网络承载了重要功能的节点易受到攻击，一个小疏忽可能酿就全网的安全问题。

在互联网流量迅猛增加的情况下，现有网络依靠对等网络（Peer to Peer，P2P）、内容分发网络（Content Delivery Network，CDN）、Web 缓存（Web cache）等数据分发技术来解决大规模数据分发与转发的问题。P2P 中，Peer 可以从其他节点获取内容。Web cache 和 CDN 中可以将请求导向有内容拷贝的邻近节点。虽然现有的内容分发机制能够在一定程度上起到加快内容分发速度的作用，但它们都是属于覆盖网络技术，底层仍然采用 IP 地址，在视频与图片等海量信息传

输的同时，存在重复传输大量内容的情况，降低了网络传输利用率，造成了带宽等资源的浪费。

为了从根本上解决互联网无法满足未来海量流量传输需求和网络传输效率低的问题，研究者们提出了新的网络体系结构，新的网络采用以内容为中心的端到网络的通信模式，对内容进行网络全局唯一命名，内容源被存储在内容服务器，路由节点根据内容名字进行路由和转发，并且可以将经过节点的内容或内容副本存储下来，传输方式由以服务器为主导的“推”改为以用户请求为主导的“拉”的方式，用户更加关心内容本身，而并不关心内容具体存储在什么位置，新的网络把内容放在了首要的位置。虽然 Web 缓存、内容分发网络、对等网络等缓存理论和技术已经得到了广泛的研究，但是面对未来网络透明化、普遍化、细粒度化的特点，这些缓存理论和模型都难以直接而无缝的解决问题。未来网络中的路由节点都可以缓存经过本节点的内容，因而，我们需要研究新的缓存存储方案来管理路由节点中的缓存内容，缓存技术缓解了数据传输造成的传输负载，使整体的网络冗余流量得以降低，同时，由于内容与所处位置分离，使得可扩展性、移动性和安全性等问题得到解决，缓存技术已经成为未来网络研究中的一项普遍而又重要的特征。

1.2 研究现状

1979 年 Nelson 提出了信息中心网络的思想[2]，Baccala 强化[3]了该思想，1999 年斯坦福大学提出了 TRIAD（Translating Relaying Internet Architecture Integrating active Directories）[4]项目。TRIAD 的理念是采用信息名字路由分组到最近副本，从而避免 DNS（Domain Name System）过程，但由于 TRIAD 将重点集中在基本文件收发上，而其他方面（如安全措施）并没有提及，且因为概念超前，在随后的两三年里，并没有重要的成果出现。

2006 年加州大学伯克利大学 RAD 实验室和网络计算机科学研究院提出了新的网络体系架构 DONA（Data-Oriented Network Architecture）[5]，改进了 TRIAD 在安全和一致性上的缺陷。DONA 使用自我认证的命名系统来命名网络中的实体（替代传统的基于 DNS 的名字系统），依靠资源解析器（Resolution Handler，RH）解析名字。DONA 采用一种树形的路由方案，当内容源被授权提供内容服务后，会向本地 RH 发送一个 REGISTER 信息，RH 收到后向注册表中添加一项

该内容对应的接口，然后 RH 会向它的父类或对等 RH 扩散 REGISTER 信息直到到达网络服务提供者（Internet Service Providers，ISPs）。用户发送 FIND 包请求内容，RH 查看自己的注册表，有与之对应的内容就会返回，若没有则向父类 RH 转发 FIND 包，直到到达 ISPs。

2010 年，美国 NSF 资助了 4 个未来互联网体系结构 FIA（Future Internet Architecture）项目[8]，分别是 NDN（Named Data Networking）[9]、MobilityFirst[10]、NEBULA[11]、XIA（eXpressive Internet Architecture）[12]。这四个项目都能解决当前互联网所面临的主要问题，但各自研究的侧重点有所相同：

NDN 项目是为了解决当今互联网端到端的通信模式。现在互联网的主要使用需求是内容的查获与传输，用户每次存取内容，都要间接映射到内容所在的物理地址，对于以获取和发布相关信息为主的互联网而言，这种通信模式略显不足。NDN 从 DONA 架构体系出发，采用内容名字进行路由，路由节点能够缓存内容，提高了内容的搜索效率，从而使数据传输更快。

MobilityFirst 项目主要是解决现有网络的移动性问题，该体系架构采用延迟容忍网络（Delay-Tolerant Networking），把移动性作为第一属性，使得网络非常自然地适应位置感知服务和环境，该项目集中在移动性、可扩展性和网络资源的公平性使用之间做出权衡，以便达到终端间的有效通信。

NEBULA 项目把云计算的数据中心作为数据计算和存储的主要场所，将更多地计算、存储和应用都搬到了“云”上，使用高速、安全和可靠地骨干网连接形成数据中心，形成了一个可以快速高效地提供计算服务的基础设施网络体系。

XIA 项目主要是解决可信通信的需求增长、网络使用的多样化以及提供网络服务的利益相关者数量不断增长，它是一个可信且可演化的网络体系架构。XIA 支持多个第一类责任者（X-centric），并且支持将来可能出现的应用场景和模式，XIA 创建了一种单一网络，在当前主要的通信实体（请求者、内容源、服务以及未来可能出现的应用）之间提供有效的通信支持。

欧洲各国也陆续开展了未来网络模型的项目研究，其中具有代表性的项目有 4WARD[13]、PSIRP/PURSUIT[14]。

PSIRP（Publish-Subscribe Internet Routing Paradigm）是由欧盟 FP7 资助、由芬兰赫尔辛基科技大学和赫尔辛基信息技术研究院等机构领衔进行的研究项目。PSIRP 以建立一个以信息为中心的发布—订阅通信模式为目的，用以取代当前以

主机 IP 地址为中心的发送—接收通信模式。PSIRP 是一个完全基于以信息为中心的概念，路由和转发机制不同于当前，采用 identifiers 标识信息，不是通过 DNS 映射到物理地址，而是通过 rendezvous 直接寻址信息。为彻底改变当前互联网体系，在 PSIRP 架构中可以取消 IP 地址。同样由欧盟 FP7 资助的 PURSUIT 项目将在 PSIRP 的基础上继续深入研究。

欧盟 FP7 资助的 4WARD 项目与 PSIRP 几乎同时开展，目标是研发新一代可靠地、互相协作的有线和无线网络技术。WP6 工作组提出了一项新的信息网络架构 NetInf（Network of Information）[6]，通过对不同类型的信息进行信息命名，对信息进行分类，并给出两种信息命名解析方案。2009 年 Park 研究中心的 Jacobson V 教授提出了以内容为中心的网络架构 CCN（Content-Centric Networking）[7]，并开展了 CCNx 项目。

互联网的快速发展和应用服务需求的多样化使得未来网络的研究出现了差异化，各个国家的科研组织也提出了各自的网络系统架构，本文我们主要研究了信息中心网络 ICN 和延迟容忍网络 DTN 这两种网络，以及两种网络架构下的缓存方案。

不同于现有网络基于 IP 地址的端到端通信方式，信息中心网络是基于内容名字的端到网络通信模式，与内容所处的位置无关[66]。此外，在信息中心网络中，用户向网络中发送包含内容名字的请求，网络通过内容名字将请求路由到有相关内容拷贝的附近节点（替换 IP 地址）。这反过来促使缓存更具有吸引力，因为使用缓存不仅可以减少延时，还能减少网络流量。近年来，关于信息中心网络中缓存的研究成为了热点。文献[15]是信息中心网络的默认缓存机制（Cache Everything Everywhere，CEE），当内容经过路由节点时就将内容缓存，是典型的均匀、非协作缓存方案，但是面临大量的缓存冗余问题。文献[16]提出了一种名为 Intra-AS Co 的缓存方案，该方案维持两张表：本地缓存表和相邻接点缓存表，通过贪婪算法旨在消除相邻接点的冗余缓存，但可能因为周期性更新表的及时性问题导致时延，而且维持两张表造成了新的开销。文献[17]提出了两种方案：LCD（Leave Copy Down）和 MCD（Move Copy Down），LCD 将流行度高的内容沿着传输路径逐步缓存到网络边缘，避免流行度低的内容占用缓存资源，但由于多个路由节点缓存同样的内容造成缓存冗余的问题。MCD 跟 LCD 相似，只是在将缓存内容沿传输路径复制到下游节点的时候，会将上游节点中的缓存删除，

从而减少了缓存冗余，但是因为它是沿着传输路径缓存内容的，在离线路径上若有新的请求，可能导致时延。文献[18]提出了 WAVE 方案，将代表内容的连续数据段分为一个个“块”，随着内容请求的增加，沿线路由节点缓存的“块”呈指数增长，该方案能够将内容热度较高的内容缓存到离用户较近的节点，在提高缓存命中率的同时降低了节点中缓存的替换开销，但它没有考虑节点的缓存能力，可能导致内容源服务器端附近出现大量的缓存拷贝，且实现复杂度高。文献[19]提出了一种选择适合的路由节点存储内容的策略，该策略通过计算路由节点的介数，将内容或内容副本存储到路径上介数最大的路由节点，但该方案未能有效利用节点介数低的路由节点的缓存空间，导致缓存分布的严重不均衡，同时可能造成介数大的节点沉重的负载，造成网络拥塞。文献[20]提出了 Probcache 策略，路由节点接收到数据包后，根据本节点距离内容源的跳数和用户距离内容源的跳数计算概率，从而决定以多大的概率缓存数据包，Probcache 综合考虑了路由节点的缓存空间大小和距离内容源的跳数，但由于每个路由节点都需要进行单独计算，且路由节点要维持 TSI 和 TSB 两个值，引入了额外的开销。文献[21]提出一种基于 Age 的协作缓存方案，该方案保证 ICN 中不同流行度的内容均匀分布，但是可能出现低流行度的内容无法被替换的问题。

延迟容忍网络是未来网络中另一种新颖的网络体系架构，主要应用在连通性频繁中断、不稳定、传输时延大和可变连通性的场景下，例如灾难救援、乡村通信、野生动物监控、军事战场等。为了解决延迟容忍网络传输链路不稳定的问题，DTN 采用“存储—携带—转发”的路由策略，路由技术是容忍延迟网络中高效传递消息的关键所在，其核心思想是从路由节点的缓存中选择合适的数据交给合适的携带者，每个节点将会携带消息副本一段较长的时间，直到遇到合适的转发节点。

由于无线节点的缓存空间大小有限，整个网络不能过多的复制同一消息副本，节点存储消息的时间有限，低效率的缓存管理机制可能会对路由算法的性能产生严重的恶劣影响，造成网络拥塞，影响网络性能，因此，当路由节点的缓存空间满时，需要一个高效率的缓存管理机制来决定丢弃哪些消息、优先转发哪些消息。

一些缓存管理机制能够独立于路由算法。文献[22]中，Zhang 等人指出在延迟容忍网络中广泛使用的传统缓存机制，例如尾部丢弃（Drop Tail，DT）和首

部丢弃（Drop Front，DF），性能表现比较差。文献[23]提出了一种基于历史丢弃的策略（History-Based Drop，HBD），HBD 在交付比率和平均时延上都胜过传统缓存策略，但它依然有很大的提升空间。

一些缓存管理策略是结合特定的路由算法来实现的。文献[24]中，Lindgren 等人基于 Prophet 路由算法，通过定义传递概率来决定优先转发哪个消息，在 Prophet 路由算法基础上对多种缓存管理策略进行了分析和性能比较。文献[25]Khelil 针对广播路由算法 Hypergossiping，在大时间尺度下基于节点间的连接提出了一套量化节点移动性的指标，通过这些指标设计缓存策略。Erramilli 等人在文献[26]中，基于 Delegation Forwarding 路由算法，通过定义的消息距离目的节点的远近，对不同的消息进行优先级高低设定，以此来设计缓存管理策略。

文献[58]通过结合游戏和定价的方式研究了一种车载网络中的协作传输。文献[59]提出了一种利用 DTN 中少数节点作为缓存的载体构成基本的协作机制，文中，路由节点竞争成为少数缓存载体，这些路由节点在网络信息传播过程中，利用自己的空间和能量进行中继操作。文献[60][61]首先选中中继节点，然后做出合适的缓存管理方案，文献[60]的主要贡献在于通过计算缓存利用值来选择中继节点，该值由节点的剩余缓存空间邻接节点决定，缓存管理算法主要通过减少缓存使用、控制流量和提供稳定数据包来分析，但是算法是基于资源树形结构多路广播路由实现的，有些特征可能并不适用所有 DTN 环境。文献[61]介绍了节点社会属性，通过计算节点的中间状态来决定它的社会集中性，并选中社会集中性高的节点作为中继节点，当节点缓存空间满时，丢弃网络中消息副本最多的缓存消息，虽然仿真结果表明，该算法有较高的交付比率，但是作者只考虑消息副本数量来管理缓存，其他社会属性对于整个网络的性能表现未能加以考虑。

1.3 主要工作和论文结构

本文分析了未来网络中的两种网络体系架构：信息中心网络和延迟容忍网络，并且在两种网络架构下讨论了各自性能比较优越的缓存方案。在信息中心网络中，分析讨论了数据块缓存位置与搜索（Caching Location and Searching，CLS）方案，通过实验仿真，在缓存命中率、缓存命中距离、时延和带宽消耗这些性能指标上，该方案均优于常用缓存策略。在延迟容忍网络中，分析讨论了基于消息传输状态的缓存替换和调度（Message Transmission Status-Based Replacement and

Scheduling，MTSBRS）方案，通过实验仿真，在交付比率、开销比率和平均时延三种性能指标上，该方案优于常用缓存策略。

本文共分为五章，各章节的内容安排如下：

第一章介绍未来网络及其缓存方案的研究背景，主要是信息中心网络和延迟容忍网络的研究现状，以及两种网络架构下的缓存方案的研究现状。

第二章主要介绍了未来网络中两种网络架构体系：信息中心网络和延迟容忍网络，分析了这两种网络架构的体系结构和工作机制，信息中心网络介绍了两种架构，分别是命名数据网络（NDN）和面向数据传输网络（DONA），然后总结了信息中心网络和延迟容忍网络架构下常用的缓存管理机制，为下文更加深入的理解和研究未来网络存储机制打下基础。

第三章重点分析了数据块缓存位置与搜索（CLS）方案，CLS 是一种隐式协作缓存管理方案，在路由节点中添加一个缓存记录，保证传输路径上同样的缓存内容只有一份，同时优化路径外请求搜索，根据当前节点到内容源的跳数和所设置的阈值的比较，决定将请求向上游还是向下游进行转发。最后，通过仿真实验，将 CLS 与常用缓存放置策略进行对比，证明了 CLS 方案的有效性。

第四章分析了基于消息传播状态的节点缓存管理方案，该方案能够抑制网络中消息副本较多且传播速度较快的消息的进一步扩散，使副本数量较少、传播速度较慢的消息得到更多地传输机会。仿真结果表明，相较于其他常见缓存替换和调度策略，基于消息传播状态的缓存管理方案能够提高路由算法的交付比率，且开销比率和平均时延均优于常见缓存替换和调度策略。

第五章总结全文，并展望了未来进一步研究的方向。

第二章 未来网络缓存方案比较研究

随着互联网承载内容的快速发展和用户需求的日益增长，针对不同的应用场景，研究者们提出了不同的未来网络体系架构，本章分析了两种未来网络架构：信息中心网络和延迟容忍网络，以及各自网络架构下的常用缓存机制。

2.1 信息中心网络体系架构

本节介绍两种不同的信息中心网络架构：命名数据网络（Named Data Networking，NDN）和面向数据的网络架构（Data-Oriented Network Architecture）。

2.1.1 命名数据网络体系架构

命名数据网络（也称为内容中心网络[27]，Content-Centric Networking，CCN）旨在用内容是什么取代内容所在位置。现在的互联网给网络中的主机/接口分配 IP 地址，而 NDN 则是使用名字对应内容，每个内容块分配一个类似于现在 URL 的分层次结构名称，格式如图 2.1 所示。这种命名风格使得用户比较容易记得名字，某种程度上也反映了内容和名字的一种映射关系。

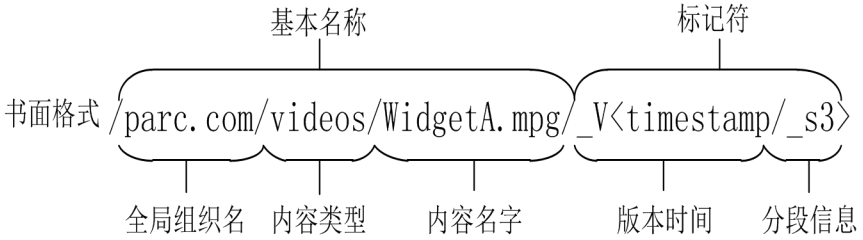

图 2.1 NDN 的命名格式

NDN 的体系架构设计主要参考了当前 IP 网络的沙漏模型，两种架构的协议栈模型对比如图 2.2 所示，细腰部分采用内容块（content chunk）取代了传统的 IP 细腰。这种架构的优势在于方便全球互联的同时，支持网络层以外各个层级的发展。

NDN 中的内容检索依靠两种类型的包：兴趣包和数据包，其格式如图 2.3 所示。兴趣包主要包含内容名字，数据包除了内容名字外，还有安全签名和数据。

典型的 NDN 节点转发模型如图 2.4 所示，包含三种表结构：待定兴趣表（Pending Interest Table，PIT）、转发信息表（Forwarding Information Base，FIB）和内容存储器（Content Store，CS）。

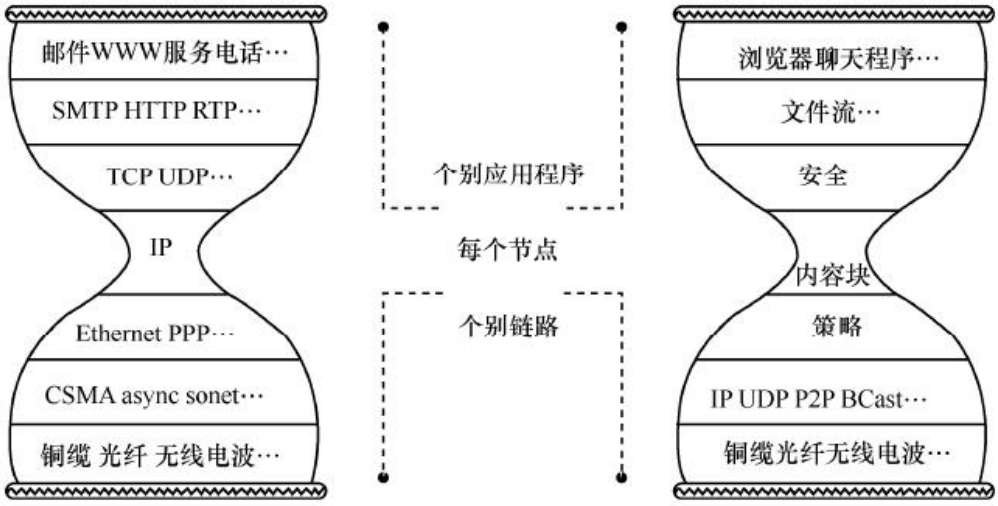

图 2.2 NDN 与 TCP/IP 网络的协议栈对比

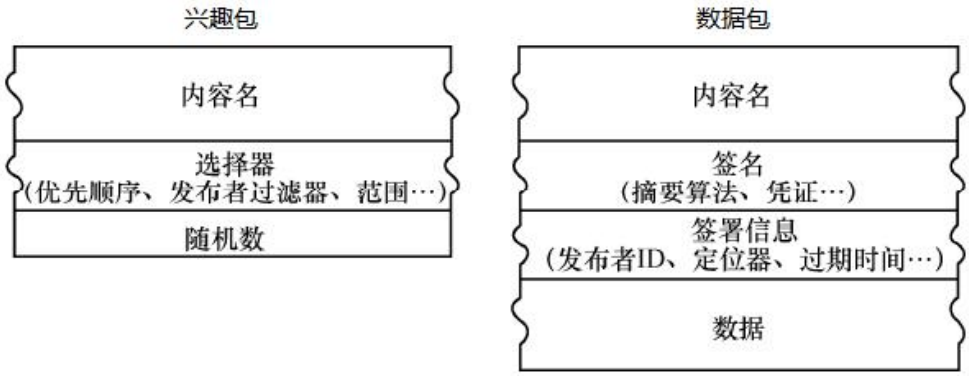

图 2.3 NDN 兴趣包和数据包格式

待定兴趣表 PIT 用于记录经过本路由节点的用户请求信息，以便响应内容能够返回发出内容请求的节点。转发信息表 FIB 有些类似于传统互联网的路由表，主要是把用户内容请求从相应的接口转发出去，与传统 TCP/IP 路由的区别在于转发信息表可以同时向多个对应接口转发用户内容请求信息，响应的数据包会从多个接口返回，显然支持多路由。内容存储器 CS 是路由节点中配置的缓存内容的存储区域，与传统 IP 网络路由器的存储和转发的区别在于，传统 IP 路由器是为了实现 IP 的转发而暂时性的缓存 IP 包，等到本次会话结束，就会清除路由器中的缓存 IP 包，而 NDN 中的路由节点可以长时间的缓存消息副本，同样的请求可以在路由节点处命中，直接返回对应的数据包而不需要再次到内容源获取信息，NDN 默认采用处处缓存（Caching Everything Everywhere，CEE）策略和最近最少使用（Least Recently Used，LRU）替换策略[65]。

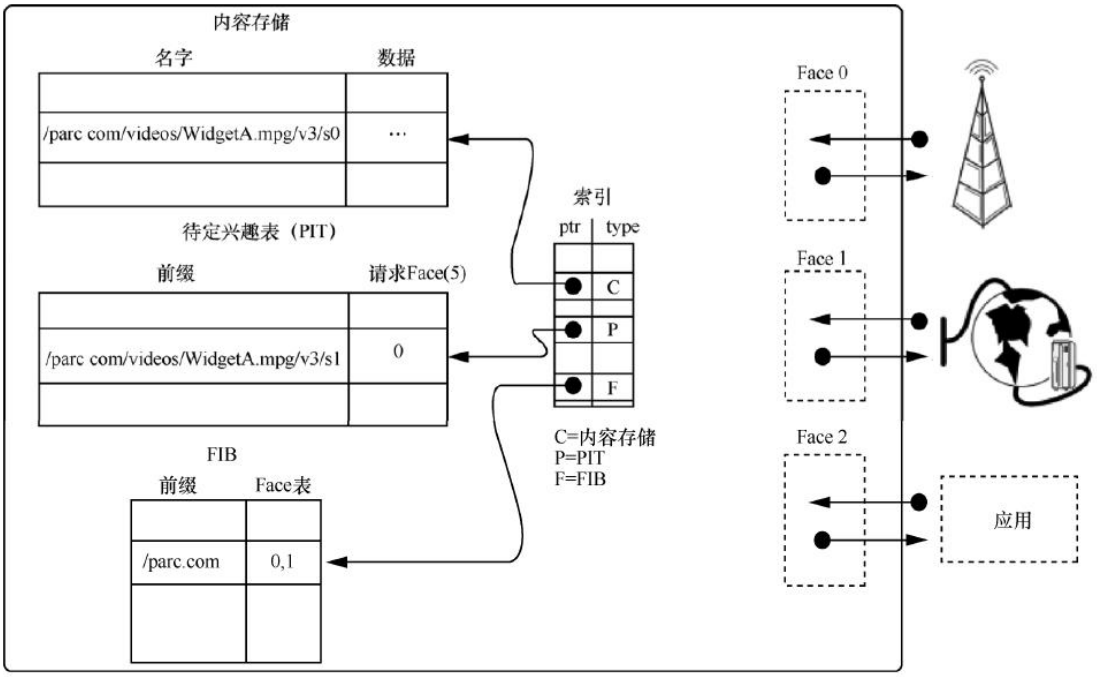

图 2.4 NDN 节点转发模型

NDN 中节点的工作机制包括对兴趣包和数据包的处理，如图 2.5 所示：

（1）对兴趣包的处理

当用户向网络发起内容请求时，他只需发送一个带有内容名字的兴趣包即可，路由节点收到兴趣包后，解包得到兴趣包中的内容名字，然后根据名字做最长前缀匹配查找，查找的表包括 CS、PIT 和 FIB，查找的优先级为：CS>PIT>FIB。若 CS 中找到匹配的内容，则直接通过收到兴趣包的接口返回数据包，然后丢弃该兴趣包，本次内容请求结束。若 CS 中没有匹配的内容，则会匹配 PIT，如果在 PIT 中找到一个匹配项，说明有其他用户发起同样的请求，该节点已经将兴趣包转发给上游节点且正在等待返回相应的数据包。此时，如果当前兴趣包与 PIT 中已有兴趣包不同，则在 PIT 中添加新的请求记录，如果两个兴趣包相同则丢弃当前兴趣包，当数据包返回时，对于 PIT 中匹配内容数据的每个接口，都会转发一份数据。若 PIT 中也没有匹配项，则路由节点会查找 FIB，若能在 FIB 中找到匹配内容，则兴趣包会被转发出去，并且更新 PIT 记录下“面包屑”以方便数据返回。如果 CS、PIT 和 FIB 都没有匹配项，NDN 会丢弃该兴趣包，表明网络中没有与之匹配的数据。

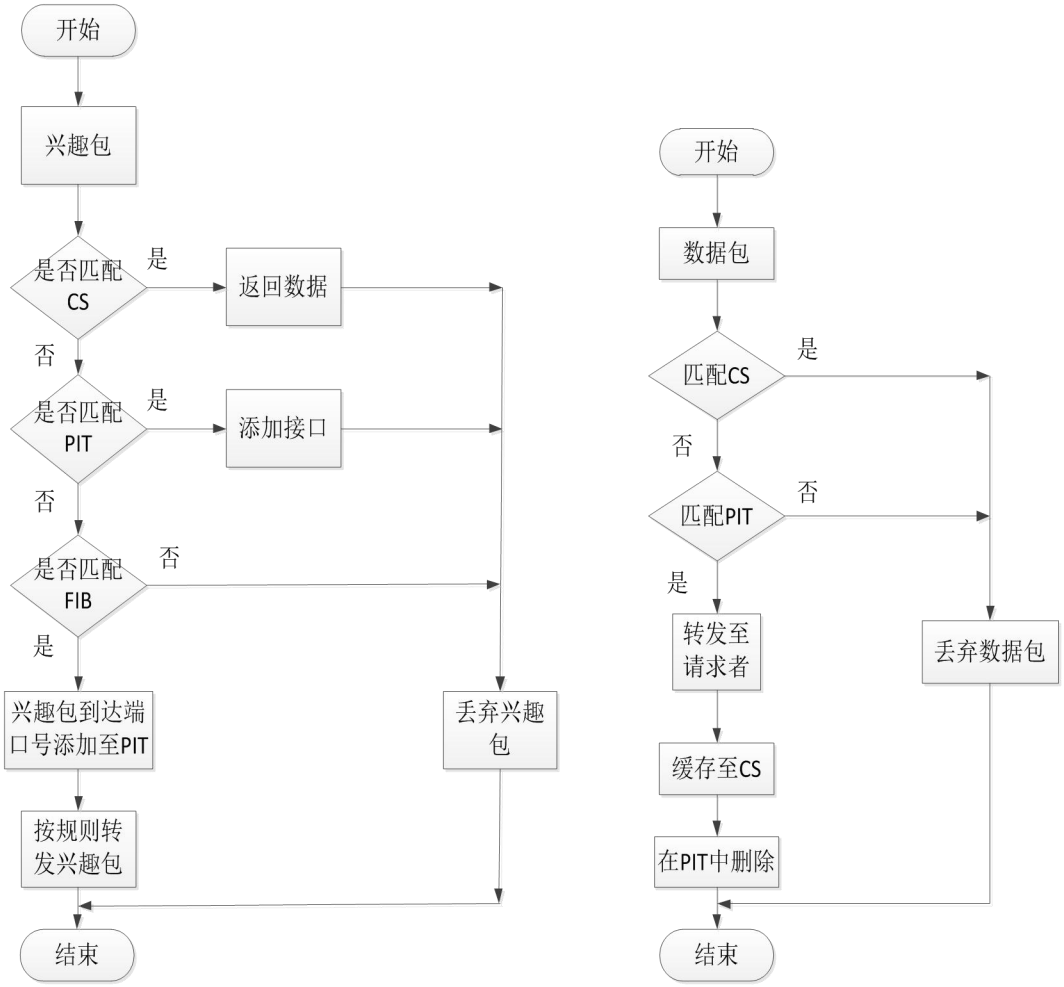

图 2.5 NDN 兴趣包和数据包处理流程

（2）对数据包的处理

数据包的处理相对而言比较简单，当路由节点接收到数据包时，解包后要对数据包中的内容名字进行最长前缀匹配。先在内容存储器 CS 中匹配，如果有相同的缓存内容副本，则丢弃该数据包；若没有，再与 PIT 中的记录信息进行匹配，如果待定兴趣表 PIT 中有匹配的记录信息，则将数据包转发给相应的接口，并将数据包缓存在内容存储器中；若 PIT 中没有匹配的记录信息，则丢弃该数据包。数据包根据 PIT 中记录的请求信息，沿着兴趣包到来的路径返回数据包。

2.1.2 面向数据传输的网络架构体系

面向数据传输的网络架构[64]（Data-Oriented Network Architecture，DONA）是另一种引起广泛关注的信息中心网络架构。为了使得内容持久唯一，DONA 采用自我认证的命名方式，形如“P:L”，其中 P 代表内容提供者的公钥哈希值，L 代表由内容提供者提供的内容所对应的特殊标签。

DONA 采用一种基于树形结构的路由方案，如图 2.6 所示。DONA 假设未来网络仍然由有着提供者/用户/对等关系的域组成，并且域中有一个资源解析器（Resolution Handler，RH）来实现内容注册和搜索功能，资源解析器在所在的域中享有同样的提供者/用户/对等关系。

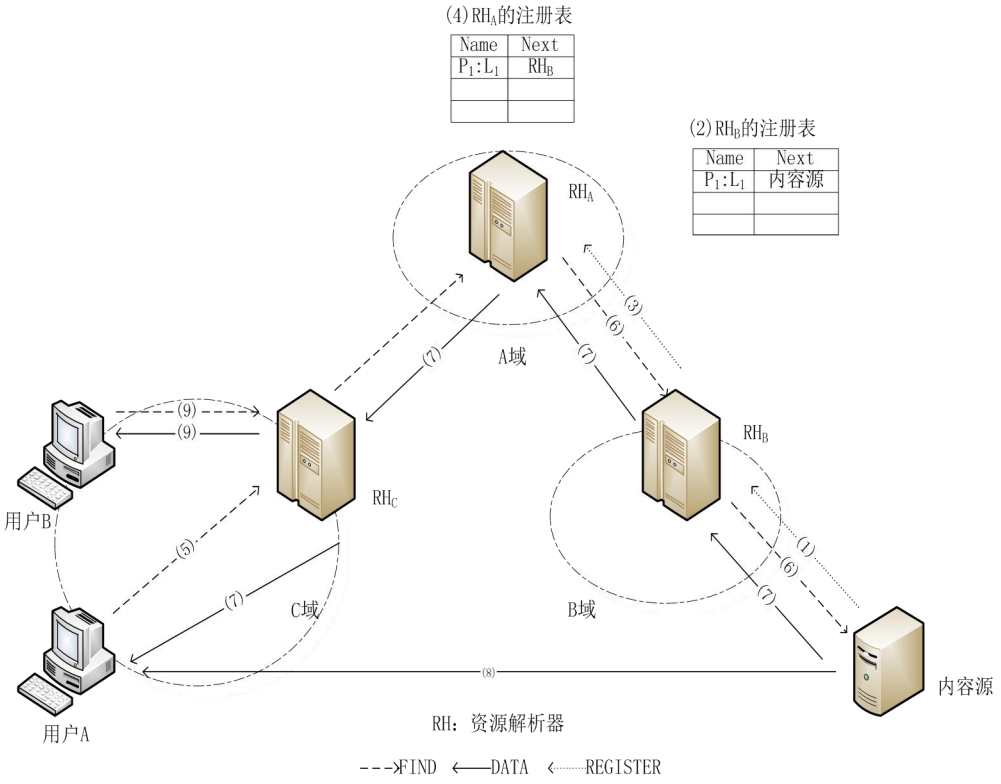

图 2.6　DONA 中内容注册与搜索

DONA 使用三种基本单元：REGISTER，FIND 和 DATA。当一个内容资源被授权提供内容服务时，它会向本地 RH 发送一个 REGISTER 信息，如图 2.6 中(1)所示。当 RH 接收到 REGISTER 信息后，它在注册表中为该内容名字添加一个接口，该接口记录了内容的名字以及受到 REGISTER 信息的 RH（或者是内容源）的 IP 地址。例如，RH_B 的注册表中名为 $P_1:L_1$ 的内容，它的接口就是内容源，如图 2.6 中(2)所示。RH 然后基于本地规则将 REGISTER 信息转发给它的父类或对等 RH，如图 2.6 中(3)。同样的，RH_A 的注册表中的接口指向 RH_B，如图 2.6 中(4)所示。当注册信息传递到网络服务提供者（Internet Service Providers，ISPs）这一层时，该过程停止。我们注意到 DONA 只登记由内容源提供的可靠信息。

当用户想要获得内容时，他只需向本地 RH 发送一个 FIND 包，如图 2.6 中(5)所示，当 RH 收到 FIND 包后，它首先检查注册表，如果表中有内容名字的接口，

RH 会将 FIND 包转发到对应的下一个路由节点。否则 RH 会基于本地规则将 FIND 包转给它的一个父类 RH，如此一来，FIND 包要么被转发给内容源，要么被 ISP 的 RH 所丢弃，如图 2.6 中(6)所示。当内容源收到 FIND 包后，它会将对应的数据包沿反向路径传递给请求者，如图 2.6 中(7)所示，或者通过一个直接路由路径传递给请求者，如图 2.6 中(8)所示。

2.2 延迟容忍网络架构

2.2.1 延迟容忍网络的特点

与传统互联网不同，延迟容忍网络的应用场景比较苛刻，主要表现在：

（一）链路特点

（a）链路可变，且延迟高。网络中节点间的通信距离、网络资源（链路和存储空间）的限制以及不对称的低数据率，都有可能造成可变的/高的延迟。

（b）连接频繁中断。无线环境中，节点的移动、节点的低功率以及障碍物的阻隔都有可能造成链路连接的中断。有些中断是可预测的，例如卫星规律性的绕地球运行；有些中断是不可预测的，例如航天器在宇宙中漫游。

（二）节点特点

（a）节点存活时间有限。在很多应用场景下，例如无线传感器网络、应急网络或军事网络，因为能源紧缺等原因，节点的寿命通常比较短。

（b）低占空比操作。当节点所处区域没有供电系统时，通常使用电池或太阳能充电，能量比较有限，所以需要限制节点的工作时间来节约能量。

（c）缓存能力受限[62]。在一些受限网络中，节点的存储能力和数据处理能力有限。当节点的存储空间耗尽时，需要有效的缓存管理机制来确保节点正常搜集数据并存储。另外，为确保可靠性实现内容重发功能，重发缓存区中的内容也会占用资源。

以上这些特点说明，传统的互联网体系结构不适用，因此需要研究适用于特殊环境的延迟容忍网络 DTN。

2.2.2 延迟容忍网络体系结构

DTN 体系结构主要的研究组织包括因特网研究任务组（IRTF）下的延迟容忍网络研究组 DTNRG[34]、以及网络研究组 IPNSIG[35]DARPA。IPNSIG 侧重于星际网络中太空远距离通信，DARPA 的研究侧重于网络出现分割时，如何改进现

有互联网应用以适用于新的网络环境。DTNRG 主要集中在源端与目的端之间没有稳定通信链路的环境中，提供实现端到端相互操作的体系架构和通信协议的设计。

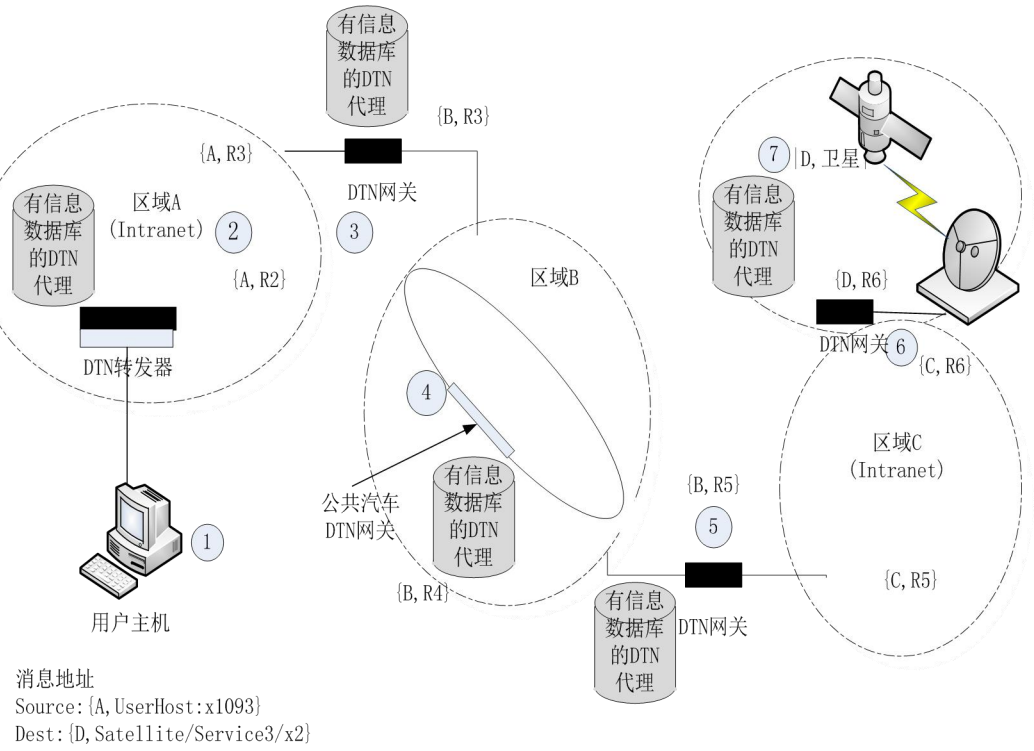

图 2.7　DTN 体系结构示意图

如图 2.7 所示，DTN 体系结构中包含了区域和 DTN 网关的概念。DTN 可以由各个不同类型的区域组成，每个区域的网络拥有网络全局唯一的标识，各个区域的网络通过 DTN 网络进行连通通信。为了进行数据的有效转发，DTN 网关对两边区域的数据进行安全性检查并可靠存储。

与传统互联网不同，DTN 的体系结构被设计来解决受限网络存在的问题。DTN 在传输层之上加了 Bundle 层，如图 2.8 所示。Bundle 层在 DTN 中起着非常重要的作用，主要包括以下几个方面：

（1）重传机制。当本节点向下游传送完数据后未收到来自下游节点的成功确认时，Bundle 层会重传，直到本节点收到下一个节点返回的成功确认信息。

（2）可以处理网络的不稳定连接。由于端到端之间并不是随时保持稳定连接，所以在 DTN 节点中设有缓存，而 Bundle 层负责协调数据的存储与转发。

（3）能够利用网络的各种连通性，包括连续的、可预测的、预定的和机会性的连通。

（4）实现覆盖网络端点标识符的晚绑定形成互联地址。

图 2.8 DTN 体系结构与传统互联网的区别

Bundle 层传输的数据包也称为消息（message），因而，DTN 体系结构是基于节点间消息互换的，传输消息的过程如图 2.9 所示。传统 TCP/IP 网络通过建立源端和目的端的路由，以数据包的形式进行消息传送，DTN 网络则是遵循“存储—携带—转发”的原则，以逐跳形式将内容机会式的传递给内容请求者。

图 2.9 DTN 体系中消息传送

2.3 信息中心网络中常见缓存策略

目前针对路由节点缓存管理的研究，一般集中在两个方面：一是缓存放置策

略，即路由节点的内容缓存器中缓存哪些内容或数据、不缓存哪些内容以及一些需要存储的内容被放置在哪些特定的节点上。目前，ICN 中的常见的缓存放置策略有 CEE（Caching Everything Everywhere）、Prob(p)、Probcache、LCD（Leave Copy Down）和 MCD（Move Copy Down），以及最近比较热门的基于内容流行度的缓存策略 MPC（Most Popular Content）等等。二是缓存替换策略，网络中的路由节点的缓存是有限的，当某一节点的缓存空间饱和时，需要将缓存存储器中的某些缓存内容替换掉，以便有足够的缓存空间存储新的内容，大多数替换策略都是基于内容的访问时间或是访问频率。目前，ICN 中常见的缓存替换策略有 LRU（Least Recently Used）、LFU（Least Frequently Used）、LRFU（Least Recently Frequently Used）等。

我们首先来分析信息中心网络中常见的缓存放置策略：

（1）处处缓存策略（Caching Everything Everywhere，CEE）

CEE 是信息中心网络中的默认缓存放置策略，用于使上游节点带宽需求减少、下游节点时延降低。如图 2.10 所示，CEE 中，请求者发出兴趣包，与兴趣包对应的数据包沿着请求者到内容响应节点的路径处处缓存，因此，CEE 是一种典型的均匀、非协作缓存方案，但是很明显，由于每个路由节点都缓存数据包的拷贝，造成了大量的冗余。

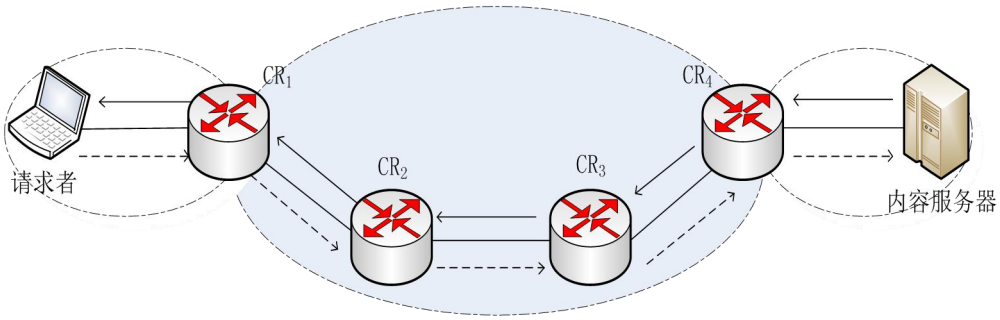

图 2.10　处处缓存策略 CEE

（2）等概率缓存 Prob(p)

Prob(p)的路由节点缓存结构与图 2.7 相似，只是当数据包经过路由节点时，并不是一定缓存，而是以概率 p 缓存，则不缓存内容数据的概率就是 1-p，p 的值由网络提前设定（可根据节点缓存容量，内容流行度，网络负荷等），当 p=1

时，与 CEE 相同。在请求者与获得兴趣包响应的路由节点间的所有节点，都独立判定是否缓存内容副本，相比于 CEE，减少了缓存冗余、提高了缓存效率。此外，当内容多次经过某个路由节点时，该路由节点缓存此内容的可能性将增强，由此看来，Prob(p)倾向于缓存流行度高的内容。

（3）概率缓存策略 Probcache

Probcache 是另外一种概率缓存机制[28]，路由结构如图 2.11 所示，它在兴趣包头部增加了启动时间 TSI（Time Since Inception），在数据包的头部增加了 TSI 和诞生时间 TSB（Time Since Birth）。用户发出兴趣包后，TSI 值设为 0。当一个路由节点接收到兴趣包后，兴趣包中 TSI 的值加 1。同样的，当一个内容源发出数据包时，它将 TSB 设为 0，TSI 的值与对应兴趣包的 TSI 值相同。当路由节点收到数据包后，将数据包中的 TSB 值减 1，TSI 不变。

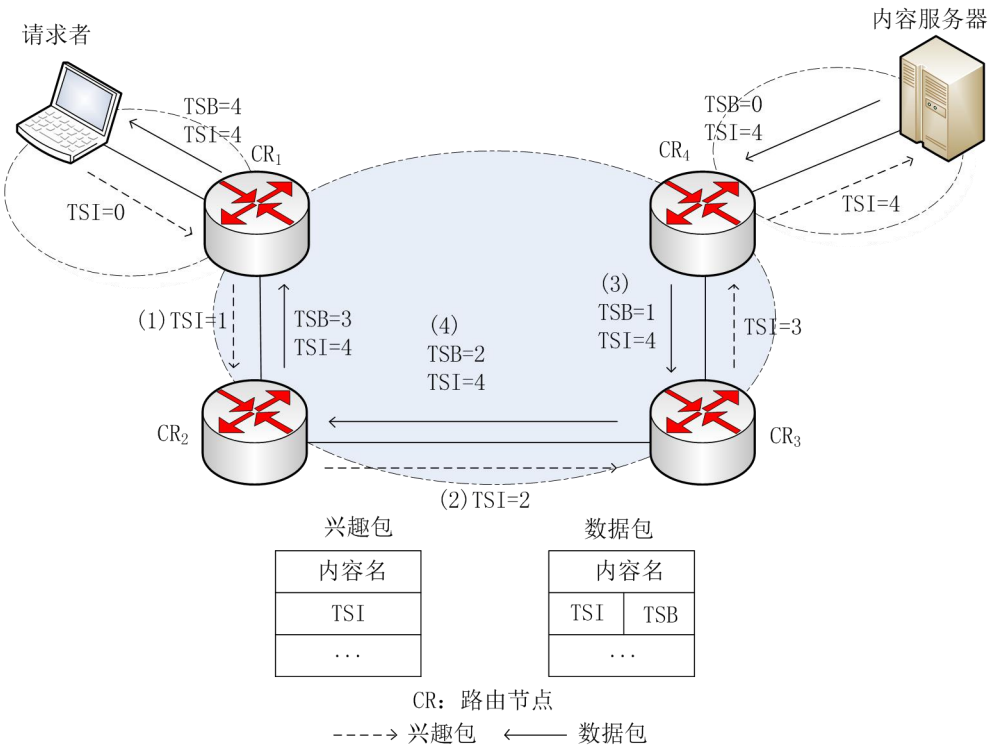

图 2.11　概率缓存策略 Probcache

Probcache 使用 TimesIn 值来评估下游节点路由中还有多少可用缓存，当一个给定的路由节点 CR_x 接收到一个数据包，它使用公式 2.1 来计算 TimesIn。

$$\mathrm{TimesIn}(x)=\frac{\sum_{i=1}^{c-(x-1)}N_i}{\mathrm{T}_{tw}N_x} \tag{2.1}$$

c 和 x 分别代表数据包中的 TSI 和 TSB 的值。T_{tw} 的值是定义的目标时间窗口，在文献[28]中 T_{tw} 的值为 10。N_i 代表传递路径中路由节点的平均缓存能力，N_x 代表 CR_X 的缓存能力。

当数据包到达内容路由节点时，利用 TSI 和 TSB 计算权重值，如公式 2.2。

$$\text{CacheWeight}(x)=\frac{TSB(x)}{TSI(x)} \tag{2.2}$$

例如图 2.11 中，当 CR_4 收到数据包，TSB 值加 1，TSI 的值仍是 4，则 CacheWeight 的值为 1/4=0.25。同样的，CR_3 收到数据包，TSB 的值变为 2，CacheWeight 等于 0.5。可以明显看出，数据包越接近请求者，CacheWeight 的值越大。

上面已然得到了 TimesIn(x)和 CacheWeight(x)的值，那么一个路由节点缓存一个到来的数据包的概率 Probe(x)，如公式 2.3 所示。

$$\text{Probe}(x)=TimesIn(x)\times CacheWeight(x) \tag{2.3}$$

同 Prob(p)相似，Probcache 也旨在减少缓存冗余、倾向于缓存流行度高的内容，但与之不同的，Probcache 是协作式的缓存方案。

（4）向下拷贝保留策略 LCD

LCD 是一种新的非均匀、协作式缓存方案[29]，旨在减少缓存冗余。LCD 中，在数据包的头部添加了一个缓存建议标志位。当用户请求到达内容服务器或者有对应请求的内容副本的路由节点时，标志位被激活，下游节点根据这个标志位来决定是否缓存内容。

如图 2.12 所示，当用户 A 想要获得存储在源端 Source 中的内容，A 向网络中发出一个兴趣包，当内容源收到兴趣包后，它返回数据包并将缓存建议标志位置为 1。当第一个下游路由节点 CR_3 收到数据包后，它发现标志位是激活的（为 1 代表激活），于是将内容缓存，同时将缓存建议标识为置为 0 并将数据包转发给下一跳路由节点 CR_2。CR_2 收到数据包后，发现数据包头部的缓存建议标志位为 0，所以就不缓存内容，同样的，CR_1 收到数据包也不缓存内容，将内容转发给用户 A，本次通信过程结束。

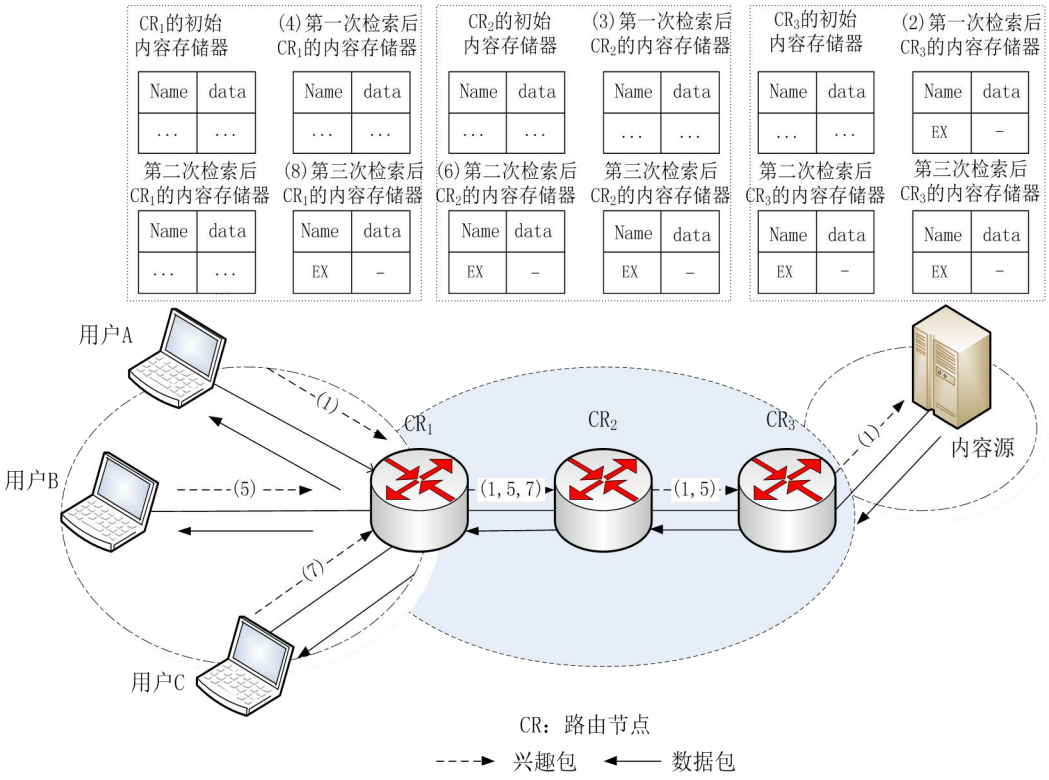

图 2.12　向下拷贝保留策略 LCD

当另一个用户 B 也想获得同样的内容时，他也想网络中发出一个兴趣包，此时，CR_3 中有与兴趣包对应的内容副本，于是 CR_3 响应本次请求，将缓存标志位置为 1 并将数据发给 CR_2。CR_2 收到数据包，发现缓存建议标志位为 1，于是缓存内容，并将标志位置为 0 转发给 CR_1，CR_1 将内容转发给用户 B。

从上述过程可以看出，LCD 能够将流行度高的内容逐步缓存到网络边缘，而阻止流行度低的内容占用缓存资源，但是同样的内容可能大量存在于整个网络中，造成了缓存冗余。

（5）向下拷贝删除策略 MCD

MCD[30]的提出是针对 LCD 中缓存冗余的问题，两者工作机制很像，区别在于，MCD 将数据内容传递给下游节点后，删除本地的缓存拷贝，如此一来，在请求者和内容源之间的传输路径上只会有一个路由节点缓存内容，这减少了网络中的缓存冗余。但是这样做，可能导致新的问题，例如，如果图 2.9 中，当 CR_2 将数据包传送给 CR_1，并且删除本地缓存后，此时有另一个用户 D，向 CR_2 发出同样的兴趣包请求，则该请求就会被传递给内容源，这就又造成了缓存未命中的新问题。

接下来，我们将分析几种常见的缓存替代方案：

（1）最近最少使用策略 LRU

最近最少使用策略 LRU[31]（Least Recently Used），这是一种在网络缓存中应用广泛的缓存替换策略，基本思想是将缓存存储器中最近最少使用的内容拷贝替换掉。通过使用一个栈结构，将最近被访问的内容置于栈顶，其他数据向栈底移动，当内容缓存存储器满时，则将栈底的内容删除。这种算法实现简单，额外开销小，所以在实际网络中得到广泛应用，尤其适用于短时间内有大量数据访问的情况下，但是当数据访问是有序扫描时则性能会很差，若是缓存空间的大小小于访问内容的大小则该策略完全失效。

（2）最少访问频率策略 LFU

LFU[32]（Least Frequently Used）的基本思想是把一段时间内访问次数最少的内容替换掉，这种策略在数据访问比较分散的情况下性能表现较好，但是在数据访问随时间变化的情况下效率比较低。

（3）最近最少访问频率策略 LRFU

LRFU[33]（Least Recently Frequently Used）同时考虑了上面所述的时间和频率两种因素，该方案为缓存空间中每一份缓存内容设置了一个变量 a，这个变量综合考虑了时间和频率的影响，表示预测缓存内容未来会被访问的概率。概率越大则 a 越大，反之，概率越小，a 的值越小。系统维护了一个最小生成堆，根节点是 a 值最小的缓存内容，每次需要进行缓存替换时，就会替换掉根节点的内容，同时为新添加的缓存设置变量 a 并加入到生成堆中。这种策略综合考量了时间和频率的影响，但由于 a 的值是固定的参数值，无法动态适应访问模式的变化，有可能造成缓存性能下降。

另外还有几种常用的缓存替换策略，比如先进先出策略 FIFO，随机替换策略 Random 等等。

2.4 DTN 中常用的缓存策略

延迟容忍网络中的缓存管理是基于路由算法实现的，基本上分为缓存替换算法和缓存调度算法。DTN 中节点的缓存空间大小受限，长期的存储和消息过多的复制会造成严重的拥塞，给网络带来很高的开销，甚至可能影响路由算法的性能，所以当缓存空间满时，需要一个有效的缓存替换策略来决定丢弃哪些消息。

当两个节点相遇后，会将某些消息发送给中继节点，但由于带宽有限、可能的意外中断以及中继节点缓存有限，决定发送给中继节点的消息顺序非常重要，也就是缓存调度策略。

（1）缓存替换策略：

（a）DF（Drop Front）策略：优先丢弃剩余 TTL 时间最少的消息。丢弃部分比较新的缓存消息，使得这些消息多次重传，导致开销增大。

（b）DO（Drop Oldest）策略：基于 TTL（time to live）值，首先进入节点的缓存消息也就是剩余 TTL 值最小的消息将被优先丢弃。只从节点局部被动考虑消息传播数量，未考虑当前网络状态。

（c）HBD（History-Based Drop）策略：基于估计的全局信息的分布式消息丢弃机制，为优化特定的性能指标。虽然考虑了消息在网络中的数量，但没有考虑消息的扩散能力。

（d）MOFO（Most Forwarded First）策略：基于消息的转发次数，优先丢弃哪些转发次数较多的消息。

我们可以看出，以上这些缓存替换策略都是基于排队论相关知识的，缓存管理不仅包括丢弃算法，还包括调度算法。

（2）缓存调度策略：

（a）FIFO（First In First Out）策略：类似于本节点的缓存按时间排成一个队列，首先进入节点的缓存消息，将会被优先转发。

（b）Random 策略：从节点缓存中随机选取消息进行转发，在一定程度上保证消息转发的公平性，但是未考虑实际应用中消息的流行度以及时间特性等特点的影响。

（c）EBMP（Enhanced buffer management policy）策略[67]：基于消息的传播状态，将整个消息调度框架划分为简易矢量交换模型、网络状态评估模型和效用评估模型，然后通过模型的独立性计算消息的效用值，当缓存满时，效用值低的消息将被丢弃。

（d）LPS（Less Probable Sprayed）和 LPF（Least Recently Forwarded）策略[68]：基于消息属性，主要是利用评估的消息副本数量和转发信息来计算消息的副本数量，然后获得传递效用功能和延迟效用功能，丢弃效用功能值小的消息。

2.5 本章小结

本章主要介绍了未来网络中两种网络架构体系：信息中心网络和延迟容忍网络，分析了这两种网络架构的体系结构和工作机制，信息中心网络介绍了两种架构，分别是命名数据网络（NDN）和面向数据传输的网络（DONA），然后总结了信息中心网络和延迟容忍网络架构下常用的缓存管理机制，为下文更加深入的理解和研究未来网络存储机制打下基础。

第三章　信息中心网络下数据缓存位置与搜索方案

3.1 引言

随着以信息为中心的服务的快速发展，一些以内容为要素的体系被提出用以满足需求，如 CDN，P2P 和 HTTP 代理，但只能在一定程度上解决问题。为彻底解决用户需求，近年研究者们聚焦于未来网络新的体系架构的研究（例如美国的 NSF GENI）。研究者普遍认同命名数据（取代物理地址）是未来网络路由的核心要素。CCN（NDN）是一种受到广泛关注的体系架构，通过在网络层推行内容存储和传输彻底改变数据传输方式。

数据块层次的缓存是 CCN 体系的一个独特特征，在系统性能上起着基础作用。近来，研究者们提出了两种数据块层次的缓存分析模型，通过使用马尔可夫链[37]和马尔可夫调制速率过程[38]评估 CCN 中的数据传输。文献[39]提出了一种带宽和存储分享的分析模型，在平衡用户性能和有限网络资源。此外，文献[40]介绍了一些 CCN 中存储管理的实验仿真评估。上述相关工作都没有探索最优缓存管理策略，事实上，将数据块动态放置在适合的中间节点上是一项重要且具有挑战性的任务。

最优缓存管理算法需要沿路径上的节点间的协作，可以分为隐式协作和显式协作。通过显示协作，节点间分享他们的状态和一些额外信息[41][42]，利用这些信息，节点决定缓存什么内容、什么时候缓存以及丢弃哪些缓存内容。但是这些显式协作方案的最大耗费是用于协作的额外信息的开销。而隐式缓存取消了如此详细的节点间通告协议，取而代之的是，利用节点本身的管理策略和每个节点的相关位置，来获得好的性能表现[43][44][45]，因此，隐式协作比显式协作更加适用。文献[44][45]中的隐式协作方案仍然有一些问题，例如，将数据内容缓存在少量中间节点中可能会造成其它终端获取内容困难。

本章我们重点分析 CCN 分层体系下一种名为数据缓存位置与搜索（Caching Location and Searching，CLS）方案的隐式协作，在服务端和用户直接连接的路由节点间最多只有一份内容拷贝，当请求来临时将缓存存储在下游节点，当缓存满时，将缓存退回上游节点。因而，在整个 CCN 网络中能够存储更多的内容，提高网络性能。此外，通过创建缓存记录来直接指导接下来的请求以便更好地检索内容块。仿真结果表明，CLS 优于现行的算法。

3.2 数据块缓存位置与搜索策略

本节我们将仔细描述 CCN 分层体系架构下的 CLS 方案。

3.2.1 CLS 基本操作步骤

CLS 的核心思想是：若在 l 层命中缓存内容，则将缓存传递给下游路由节点 $l-1$ 层；若是 l 层因为缓存空间满而丢弃缓存时，则将被丢弃的缓存内容上传给上游节点。同时，沿着传输路径建立一个缓存记录来帮助内容搜索，详细描述如下：

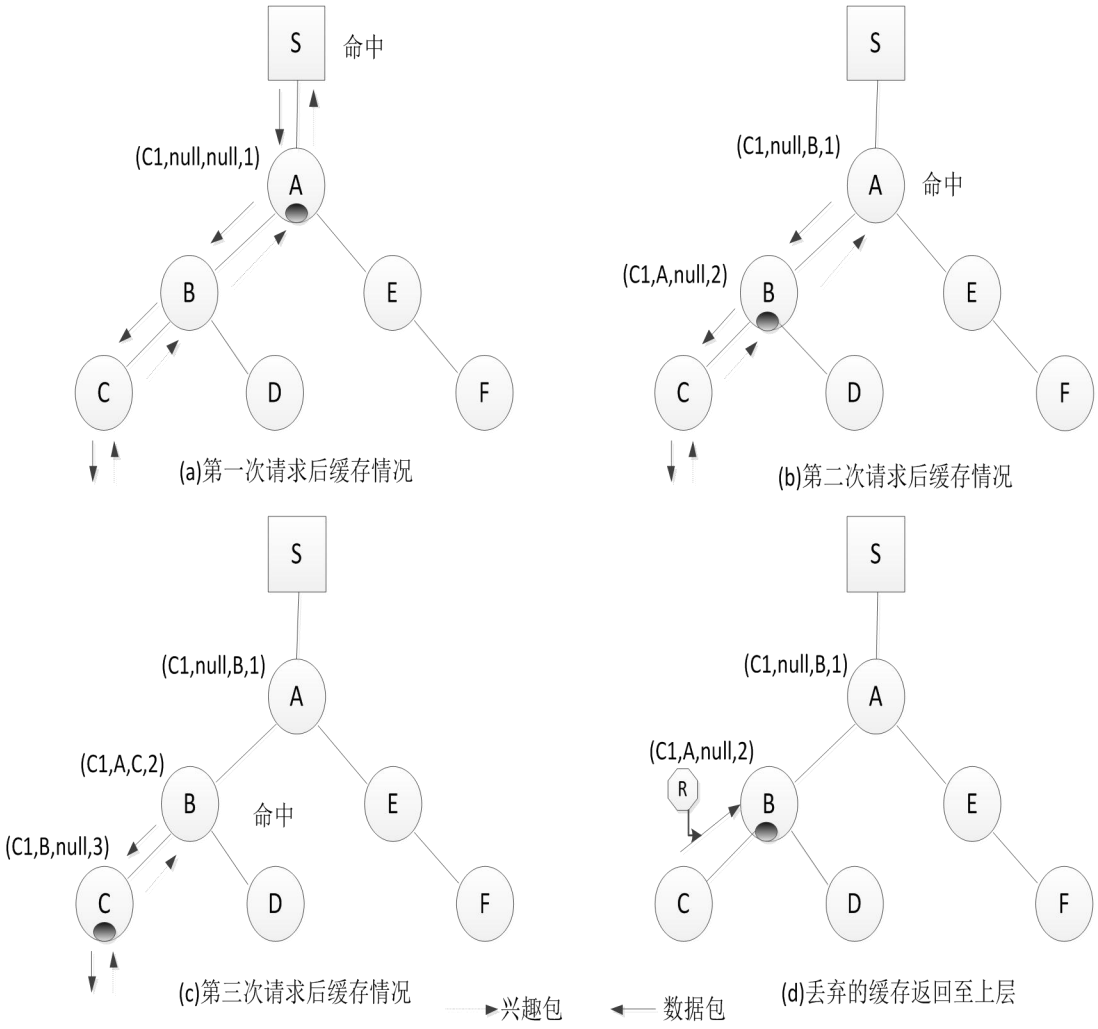

图 3.1　CLS 方案操作步骤

（1）向下缓存：在 CLS 中，若在 l 命中缓存则将缓存存储在 $l-1$ 层节点的缓存存储器中，并将 l 层中对应请求的缓存替换（移至栈底），同 MCD 的操作一样。当然，若缓存命中在服务端，则不会产生删除操作。CLS 需要通过多次请求才能将缓存带到叶子路由节点（参考树形结构的叶子定义），每一次请求使得缓

存向用户终端更近一跳，如图 3.1(a)到(c)所示。因此，在服务端和叶子路由节点的路径上，最多只有一个缓存内容。向下缓存背后的思想是防止替换错误的增大，以及获得缓存独占性（即服务端到叶子路由节点同样的缓存内容只有一个）。

（2）被替换的缓存返回至上层节点：如图 3.1(d)所示，当 l 层的内容存储器根据某种替换策略（例如使用 LRU）丢弃缓存时，CLS 方案会将缓存内容贴上标识 R 并返回到上层节点，标识 R 稍后解释。因此，一个请求数据块可能会被一直往上游节点返回，直到服务器。除了返回至服务端，CLS 保证在服务端和叶子路由节点间永远只有一份内容拷贝。所以，确保服务端和叶子路由节点间只有一个缓存内容，能够指导兴趣包至缓存路由节点而不是服务端，因而减轻了服务端的工作负荷、减少了时延。

（3）通过缓存记录进行搜索：CLS 中传输路径上的每个路由节点都建立一个缓存记录。每个记录有四组参数（ID，in，out，h），其中 ID 是内容的全局唯一 ID，记录包含以下信息：（a）in 代表内容来自哪个路由节点；（b）out 代表本节点将要把内容缓存传递出去的下游节点；（c）h 代表本路由节点到服务端的跳数。每个路由节点都修改 PIT，目的是存储之前的缓存内容的信息。

CLS 中缓存记录与 Breadcrumbs 的主要区别在于，CLS 中的缓存记录是在内容被节点缓存后建立的，而不是像 Breadcrumbs 从节点经过就建立记录。图 3.1 展示了缓存记录创建的过程：当第一次请求使得路由节点 A 缓存内容时，路由节点 A 创立一个记录(C1，null，null，1)，如图 3.1(a)。in 的值为 null 意味着内容来自服务器，out 的值为 null 代表内容缓存在本地而不需要缓存到下游节点。h 的值由数据块携带，每向前传递一跳数值加 1。第二次发起请求时，缓存向下传递到 B，则 A 的缓存记录修改为(C1，null，B，1)，创建 B 的缓存记录为(C1，A，null，2)（如图 3.1(b)所示）。因此，第三次请求后，A、B、C 的缓存记录如图 3.1(c)。另外，如果缓存内容因为替换而被丢弃，则缓存记录也会被删除（如图 3.1(d)），这种情况下，当缓存了带有标识 R 的内容后，B 的缓存记录也将修改为(C1，A，null，2)。

所以，路由节点中缓存信息的记录意味着有该缓存存储在本节点或下游节点，这可以用来指导兴趣包的搜索。一个明显的问题是搜索内容的方向：是向上到服务端，还是向下至内容拷贝，我们通过比较带有事先设定好的阈值（H_{th}）的路由跳数来制定了一个简单规则。根据 ISP 的实际环境，将 H_{th} 的值设为从服

务端到叶子路由节点的总跳数的一半是合理的，这已经超出了本文的范畴就不予以讨论了。然后，根据缓存记录信息，当且仅当 h 的值大于等于 H_{th} 时，中间节点会将兴趣包向下游转发，而不是根据 FIB 向上游转发。图 3.2 展示了搜索规则的例子，H_{th} 设置为 2。结果是，为了搜索同样的数据块（C1），路由节点 B 将来自路由节点 D 的请求向下游转发（如图 3.2(a)），而路由节点 A 将来自路由节点 F 的请求向上游转发（如图 3.2(b)）。

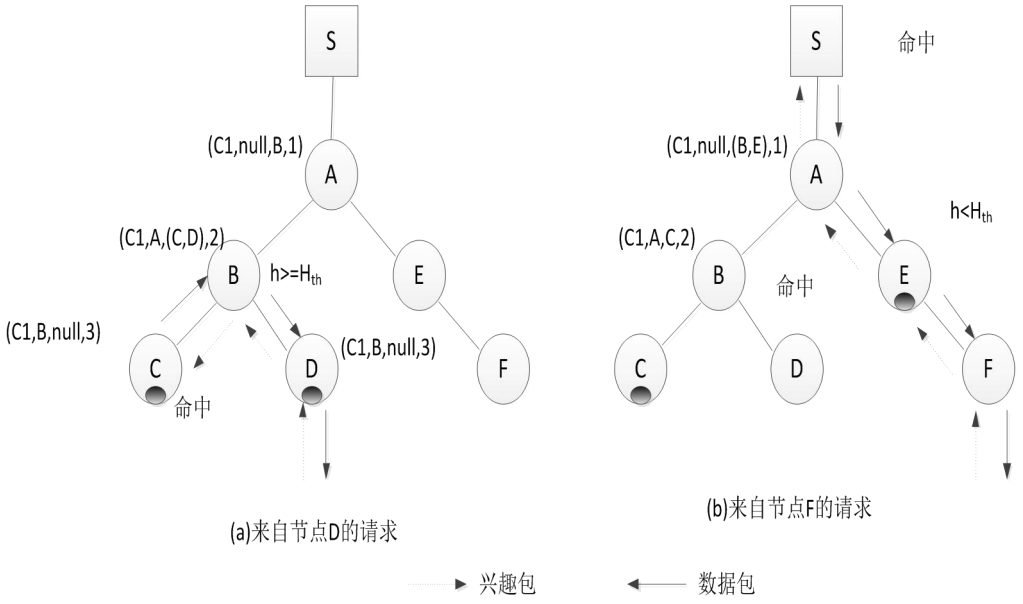

图 3.2　CLS 中缓存记录的创立过程

很明显，CLS 优于 Breadcrumbs，因为 CLS 能够保证，当向下转发兴趣包时能够搜索到缓存内容，同时通过按时删除缓存记录能够节省缓存空间。

3.2.2 CLS 方案中的特殊场景

此外，为了保证 CLS 能够正确工作，还有一些特殊情况需要解释，为了实现缓存独占性，CLS 要求在服务端到叶子路由节点的传输路径上只有一份数据拷贝，这是为如下特殊情况设计的标准。在详细描述之前，我们首先阐述一下标识 R。在 CLS 中，每份数据被添加了一个标识 R，用来鉴定数据块的状态，更准确的说，当数据块被命中时，R 值为 1；当数据被缓存时，R 值被修改为 0；当缓存数据被丢弃时，R 值设置为 2，如图 3.2 所示。

当有缓存记录的路由节点接收到一个标识 $R=1$ 的数据，特殊情况产生了，与上面所述的向下缓存不同的是这里没有缓存而是直接将数据块发出去，如图 3.3 右下角部分。图 3.2(a)举了一个这样的例子，路由节点 B 将命中数据发给节点

D 而没有缓存，确保路径 S-A-B-C 和路径 S-A-B-D 只有一份缓存，而节点 B 也需要更新缓存记录，在 out 中添加 D，h 值也是缓存记录中的 h 值和从接收数据块得到的 h 值，两者的较小值（即 $h=\min(h_{trail},h_{chunk}+1)$）。正如这个例子所说的，当节点 B 从节点 C 收到一个 $h_{chunk}=3$ 的数据块，节点 B 本身的缓存记录 $h_{trail}=2$，则将节点 B 的缓存记录中 h 值和数据块中的 h 值设为 2。更进一步，应该指出当命中数据根据缓存记录被传递到之前的到来接口 in 时，命中数据不会被从内容缓存器中删除，如图 3.2(a)。

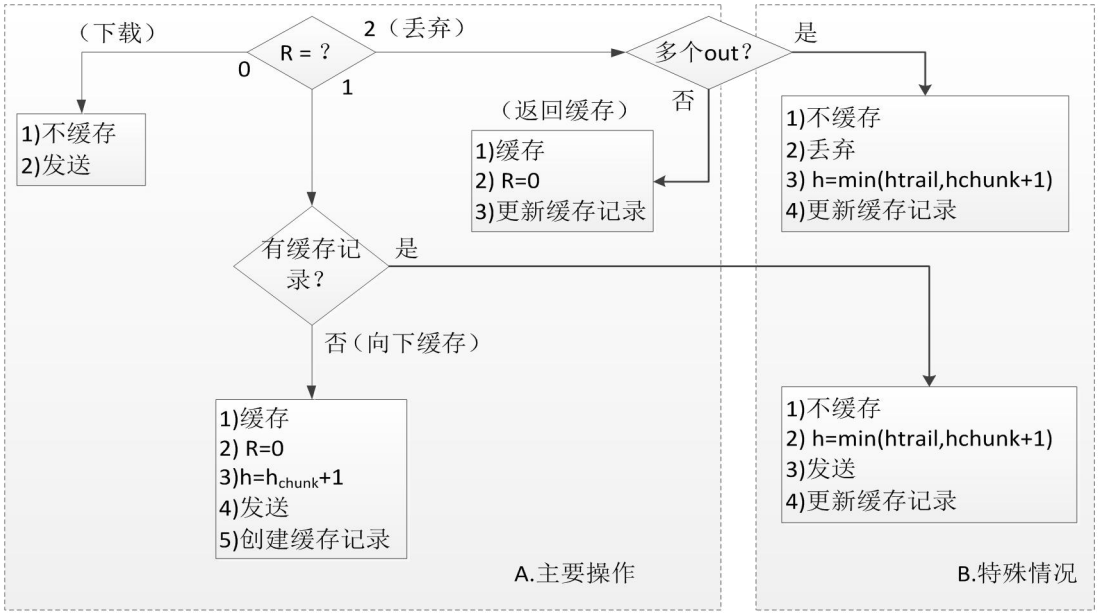

图 3.3　CLS 中路由节点接收到数据后的操作

当被替换的缓存数据被传回有多个 out 值缓存记录的路由节点时，另一种特殊情况将会产生，这种情况的操作不同于上述缓存回传，如图 3.3 右上角所示。如图 3.4，节点 B 的缓存记录中 out 值表明在经过 B 的一条路径上只有一份数据缓存，因此节点 B 删除替换数据（R=2），结果节点 B 更新缓存记录删除 out 值中的 C。

3.3 对比策略

为了分析 CLS 方案的性能指标，我们将从缓存命中率、缓存命中距离、时延和带宽消耗这些性能指标上，同一些常用的缓存存储方案进行对比分析，本节我们将简要介绍 CEE、Prob(p)、Probcache、LCD、MCD 这些策略。

（1）CEE

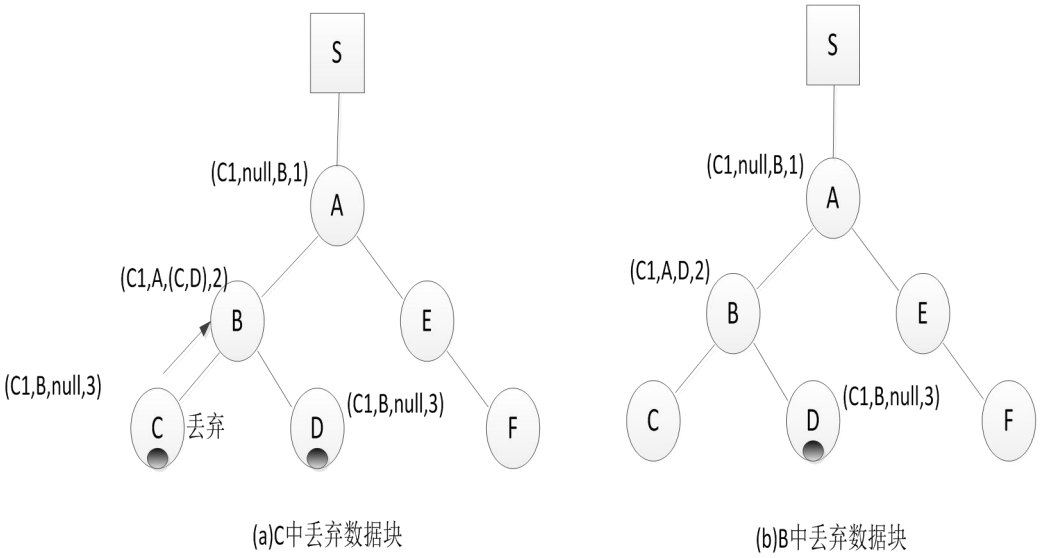

图 3.4　当节点 B 有两个下游节点时删除缓存的操作

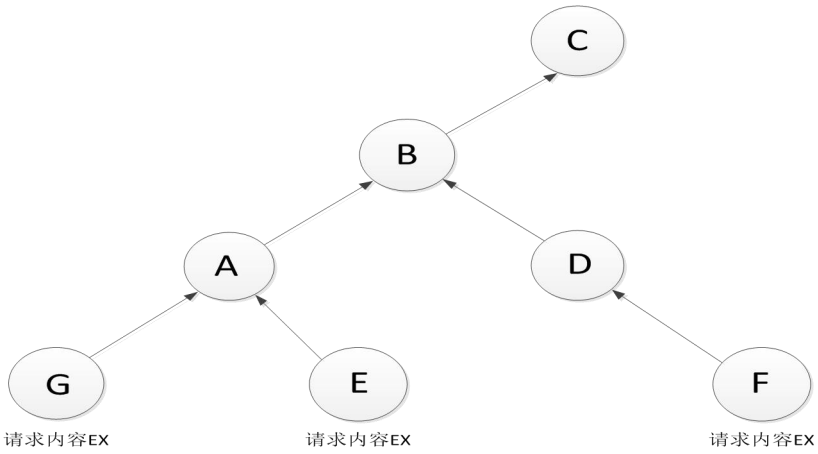

图 3.5　CEE 缓存示意图

内容会被缓存在传输路径上的所有路由节点。如图 3.5 所示，当节点 G 从服务器获取内容 EX 后，EX 将被缓存在节点 A 和节点 B，当节点 E 也要获取内容 EX 时，将在节点 A 命中所需内容；当节点 F 也想获取内容 EX 时，将在节点 B 命中，同时命中内容也将缓存在节点 D 中。

（2）Prob(p)和 Probcache

如图 3.6(a)，在用户请求与服务端的路径上，所有路由节点以概率 p 独立决定是否缓存数据，p 的值可以根据网络的实际情况（如缓存容量，内容流行度和网络负荷等），特殊的是，当 p=1 时，等同于 CEE。

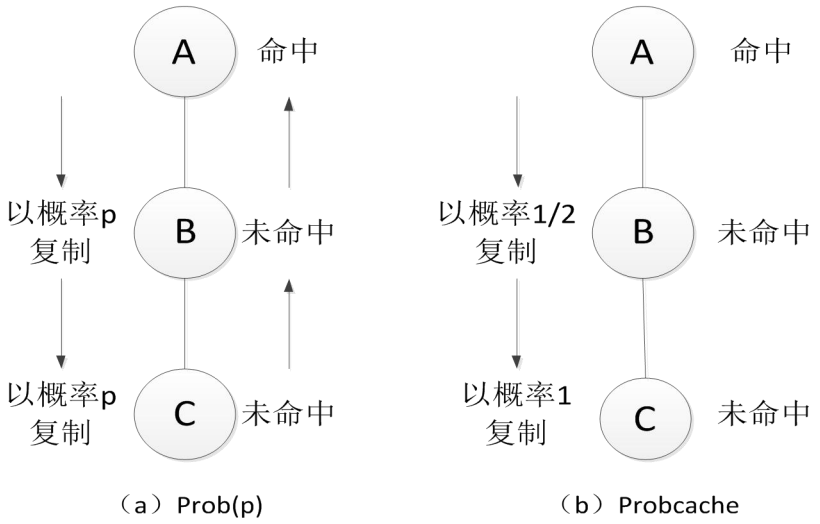

图 3.6　Prob(p)和 Probcache 缓存示意图

如图 3.6(b)，节点缓存数据的概率与节点与当前节点离服务端的跳数有关，以用户到服务器的总跳数为分母，以当前节点到服务端的跳数为分子，显然越靠近用户侧，节点缓存数据的概率越大。

（3）LCD 和 MCD

如图 3.7(a)所示，当请求在节点 A 命中时，只将缓存存储在节点 A 的下游节点 B 中，这样可以避免中间节点过多的缓存同样的内容，经过多次同样的请求，缓存内容能够到达网络边缘，靠近用户终端。

如图 3.7(b)所示，当请求在节点 A 命中时，节点 A 将缓存内容传递给节点 B 并删除本身的缓存，只保留节点 B 中同样的缓存，经过多次同样的请求，缓存内容被传递到网络边缘，靠近用户终端。同 LCD 相比，MCD 进一步减少了网络冗余，在将缓存传递给下游节点的同时，删除本身同样的缓存。这对于路径上请求非常适用，但对于路径外请求，会造成缓存未命中、时延增加的现象。

3.4 仿真实验

为验证本章分析的数据块缓存位置与搜索方案的性能，本节通过 ccnSim 仿真工具实现对 CLS、CEE、Prob(p)、Probcache、LCD、和 MCD 缓存策略的仿真，在缓存命中率、缓存命中距离、平均下载时延和带宽消耗这些性能指标上进行了分析对比，仿真结果表明，CLS 方案优于上述常用缓存方案。

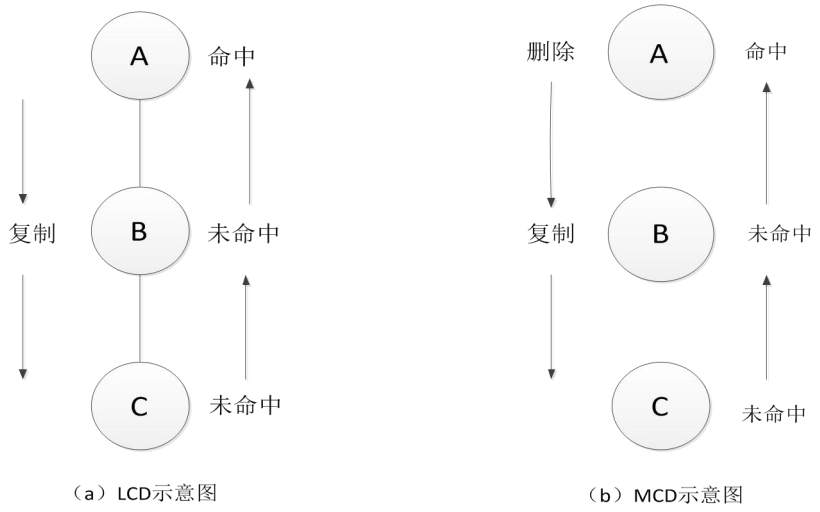

图 3.7　LCD 和 MCD 缓存示意图

3.4.1 仿真工具

本文采用 CCN 仿真器 ccnSim[47]，ccnSim 是大规模 Chunk 级别的仿真模拟器，采用 C++编写的基于 Omnet 框架。ccnSim 提供多种应用在内容中心网络的缓存管理方案，业务产生器提供 3 种业务模型：MZipf 分布、Zipf 分布和 flash crowed 模型。网络拓扑产生器提供 Cascading、abilene 和 Binary Tree 等拓扑结构。转发策略提供最短路径优先、最近复制路由等路由策略。接下来我们将从拓扑定义、内容处理、内容存储和决策策略、客户端、输出几个方面介绍 ccnSim。

拓扑定义：

"network"在 ccnsim 仿真中代表最高等级的模块，在这里面，使用者可以定义不同 CCN 节点模块间的联系、cilent 和 repositories 的布置、网络中使用的节点数量。每个网络模块，都必须继承 base_network。

Clients placement

Clients 代表用户的集合，所以至多一个活跃的 client 连接到一个给定的节点。节点通过端口 0 与 client 连接，布置还指定了有多少 client 是活跃的，client 的活跃性发生在语法 client::initialize()。

内容处理：

content_distribution 模块没参与自身的构筑，但是它为正确的运行仿真完成了很多重要的任务。

Catalog initialization

catalog 是一个 table of contents，被以下参数所描述：

objects:代表 catalog 的基数，表达的是 contents 的数目。

file_size：因为 contents 被描述像一个几何分布，这个参数代表文件大小的平均值，用 chunks 来表示。另外如果 file_size 设置为 1，每一个文件大小就是一个单独的 chunk。

replicas：这个参数表示每个内容的复制度，换句话说，第 i 个 content 将被随机完全地复制到 replicas repositories。

内容存储和决策策略：

node 模块代表了 ccnSim 的核心，它由三个下层模块组成：core_layer, strategy_layer, and content_store 模块。

Core Layer（核心层）

它实现的是 ccn 节点的基本任务：下层模块间的交流；接收 interests，返回数据（在 cs 中查找数据）；在存在的 PIT 项中附加端口；不存在时创建。

Cacheing strategies（缓存策略）

F--Forwarding strategies:决定应该使用哪些路径。

D--decision policy：说明是否路径上的当前节点应该缓存数据。返回的是 boolean 值。

R--replacement：说明缓存满了的时候什么样的元素应该被删除。

ccnSim 里给定的缓存决策算法：

DS = lce：Cache Everything Everywhere 每个地方都保存。

DS = lcd：Leave Copy Down 向下缓存。

DS = btw：在一个路径上最大的中介中心存储。

DS = fixP：固定概率缓存。

DS = Probcache：节点离副本越远，缓存概率越低。

DS = Never：禁用缓存。

DS = cost-aware P：实现 Cost-Aware(CoA)决策。缓存可能性与 price 成正比。P 是不同价值下缓存可能性的加权值。

DS = ideal_blind：实现 Ideal-Blind 决策。来的内容的等级大于等于驱逐候选项时会被接受。

DS = ideal_cost-aware：实现 Ideal-CoA 决策。来的内容的 price 大于驱逐候项 price 时会被接受。

base_cache::initialize()记录了缓存决策是怎样建立的。

最后，在节点模块里，我们可以选择缓存使用的方式，这可以通过节点复合模块的 RS 参数来实现。

RS = lru_cache：实现 LRU 替换策略，这个也是论文中常用的。简单的替换最近没有使用的项。

RS = lfu_cache：实现 LFU 替换策略，通过计算器，LFU 缓存可以识别哪些是流行度最低的内容，并在缓存满了的时候删除。

RS = random_cache：实现随机替换策略，随机选择一个元素删除。

RS = two_cache：LFU 和 Random 的扩展，随机选择两个元素，删除流行度较低的一个。

RS = fifo_cache：First In First Out。

客户端：一个 client 代表一个用户集合，它是以 Possion process（泊松过程）作为模型模拟的。在当前的实现中，client 请求文件是一块接一块的（假定 chunk window W 是 1），下列参数描述 client 的行为：

lambda：泊松过程的总到达率。每个 client 可以不同。

check_time：实现一个定时器，秒计。每次 client 模块检测下载状态。

RTT：实现网络的 Round Trip Time，如果下载文件时间超过这个值，client 假定下载已过期。并做一些事（通常是重新发送兴趣包）。

Output files

输出的是标准的 omnet++文件，特别是 coarse grained。

每个节点的统计数据收集在相应的.sca 文件里。另外 fine grained 和每个内容的统计数据被收集在.vec 文件里。

vector-recording = false

如图 3.8 展示了 ccnSim 仿真步骤图。

3.4.2 仿真环境

我们做了大量的仿真实验来对比 CLS 与常用策略的性能表现，仿真环境建立在自定义的 CCNx 拓扑[48]结构上，如图 3.1 所示。网络中有 30 个路由节点，100 个客户端和 100 个内容源，客户端和内容源随机的连接在路由节点上。此外，假设每个内容源存储 120 到 300 个数据块，每份数据块的大小为 10KB。内容源为所存储的数据块提供可用性，同时当收到兴趣包时返回数据包。

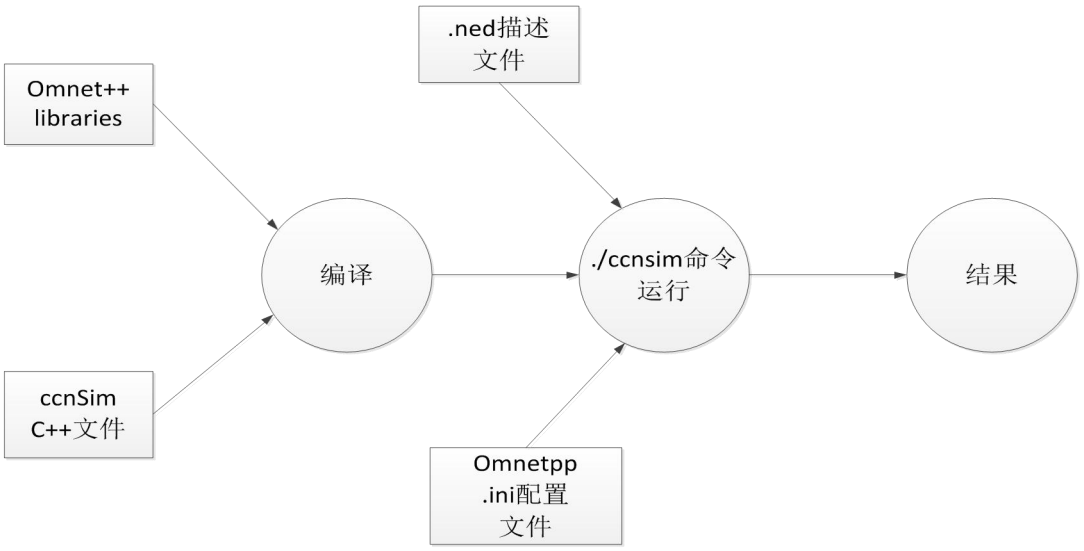

图 3.8 ccnSim 仿真步骤图

客户端发出所需数据的兴趣包，内容的流行度服从 Zipf 分布，其参数设置为 0.95。同时假设兴趣包的到达服从泊松分布，每个路由节点有同样的缓存空间。

表 3.1 仿真参数设置

网络拓扑	自定义的 CCNx 拓扑
路由节点数量	30
域的数量	1
客户端数量	100
内容源数量	100
数据块数量和大小	30000，平均 10KB

3.4.3 仿真结果

我们现在展示大量仿真的实验结果，以此来获得对网内缓存机制的刚深入理解。图 3.9 展示了不同缓存管理方案在不同的缓存空间下的缓存命中率，缓存命中率的定义是评估缓存性能和缓存内容可用性的重要指标，其计算公式如下：

cache hit ratio = cache hits/(cache hits + cache misses)

其中，cache hits 表示缓存节点响应兴趣包的次数，而非内容源响应兴趣包，cache misses 为缓存节点没有响应的兴趣包次数。

从图 3.9 中可以看出，随着缓存空间的增加，缓存命中率也在不断增加，这是因为路由节点拥有更大的缓存空间，就可以存储更多地内容，就有更大的机会能够响应兴趣包，缓存大小占比指的是缓存空间大小占所有内容的比例。相较于其他缓存策略，CLS 缓存管理策略更加能够提高缓存命中率，例如当缓存空间大

小为 3%时，CEE，Prob(0.1)，Probcache(20)，MCD，LCD，CLS 的缓存命中率分别为 28%，54%，52%，54%，59%，62%。我们注意到，MCD 的性能表现反而不如 LCD，这可能是因为 MCD 将缓存存储在下游节点而本节点删除缓存，可能导致路径外的缓存命中率降低，而 CLS 对于路径外的新的内容请求，有着很好的处理办法，因而能够显著提升缓存命中率。

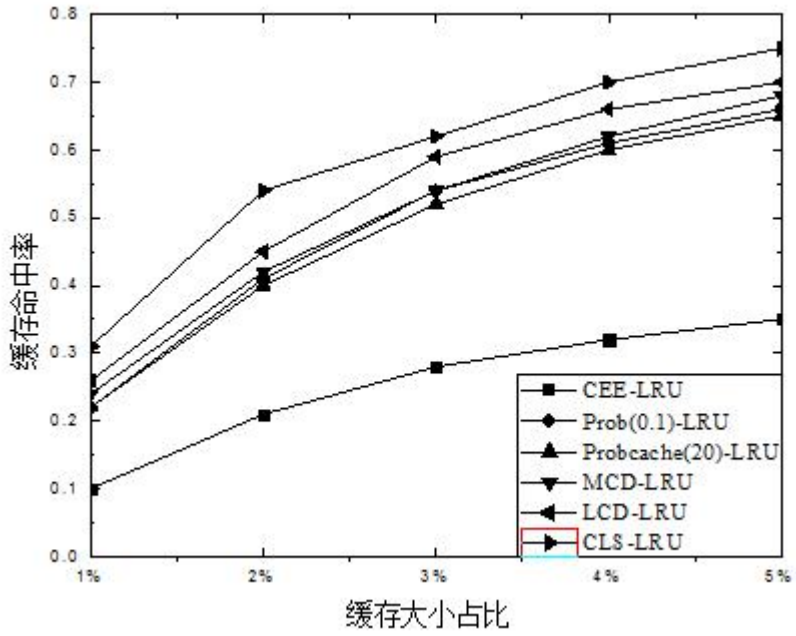

图 3.9　不同缓存策略的缓存命中率

图 3.10 展示了不同的缓存管理策略在不同的缓存空间大小下的平均命中距离，命中距离是指从用户客户端发出请求到兴趣包命中所经过的路由跳数，缓存命中距离越小，性能越佳，缓存大小占比指的是缓存空间大小占所有内容的比例。可以看出，随着缓存空间大小的增加，缓存命中距离逐渐减小，这是因为，路由节点的缓存空间增大，则能够缓存的数据副本增多，兴趣包越有可能在路由节点被命中，而非跳数更加多的内容源。我们注意到，虽然 LCD 的缓存命中率比 Prob(0.1)，Probcache 要高，但是缓存命中距离却并不比上面缓存策略少，这是因为，虽然 LCD 有能力缓存频率次数较多的内容请求，但是需要经过多次才能将内容副本缓存至网络边缘，而且缓存空间受限，可能导致传播过程中一些内容被替换，而 LCD 这种慢扩散可能被影响。CLS 方案相比于其他缓存策略，能够更加有效的减少缓存命中距离，这是因为 CLS 可以减少替换错误，并且在传输路径上同样的缓存副本只保留一个，保证了缓存的独占性。

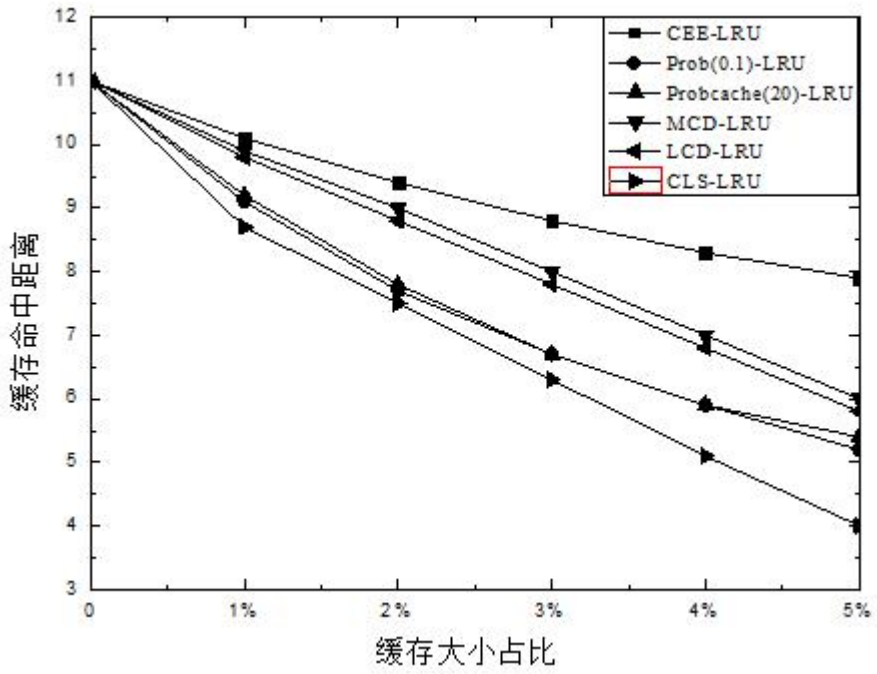

图 3.10　不同缓存策略的缓存命中距离

图 3.11 展示了下载时延，缓存大小占比指的是缓存空间大小占所有内容的比例，通过使用本章提出的搜索规则，CLS 比其他缓存策略更能够减少下载时延，CLS 的下载时延大约比 LCD 少 50ms，因为 CLS 方案可以通过比较当前节点到内容源的跳数和所设置的阈值，决定将兴趣包向上游内容源转发，还是下游节点转发，对于路径外的路由节点请求，这节省了时间。

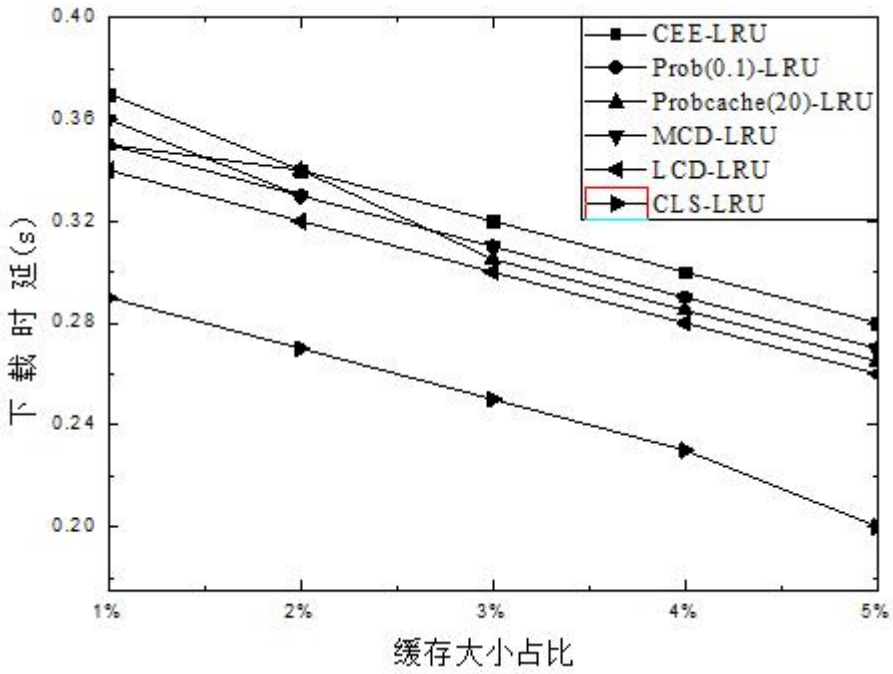

图 3.11　不同缓存策略的下载时延

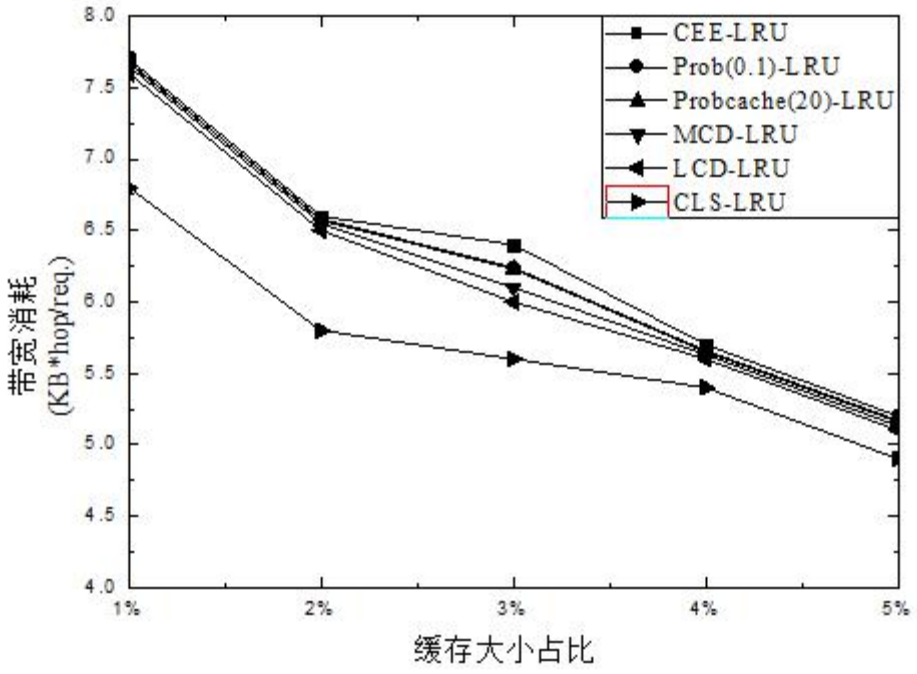

图 3.12　不同缓存策略的带宽消耗

最后，我们分析了网络中的负载情况，图 3.12 展示了平均网络流量（使用 byte*hop 度量）的情况，缓存大小占比指的是缓存空间大小占所有内容的比例，相比于其他缓存方案，CLS 方案能够显著减少网络负载，这是因为 CLS 方案，能够保证请求者到内容源的路径上，有且只有一个内容副本，当本节点缓存空间满时，则将缓存返回至上游路由节点，对于路径外请求，能够根据节点到内容源的距离和阈值的比较，决定向上游还是下游转发请求，这些措施大大减少了网络中的流量，有效降低了网络负载。

3.5 本章小结

本章重点分析了数据块缓存位置与搜索（CLS）方案，CLS 是一种隐式协作缓存管理方案，在路由节点中添加一个缓存记录，用于记录节点缓存内容副本情况，保证传输路径上同样的缓存内容只有一份，同时优化路径外请求搜索，根据当前节点到内容源的跳数和所设置的阈值的比较，决定将请求向上游还是向下游进行转发。最后，通过仿真实验，在缓存命中率、缓存命中距离、平均下载时延和带宽消耗这些性能上，将 CLS 与常用缓存放置策略进行对比，证明了 CLS 方案的有效性。

第四章 DTN 下基于缓存消息传播状态的存储方案

4.1 引言

延迟容忍网络（Delay Tolerant Network，DTN）是一种有着长时延、频繁中断等特点的新型网络体架构，例如用于野生动物跟踪的传感器网络[50]、口袋交换网络[51]和军事网络[52]等等。在 DTN 中，稳定的端到端通信可能不会总是存在，因此，在移动自组织网络中像 AODV、DSR、TORA 等传统协议，工作效率不高。在 DTN 中通常利用节点的移动性来解决端到端通信连接问题，在这些环境中采用“存储—携带—转发”的原则传输消息。

许多研究提出的 DTN 路由缓存方案都有一个前提，那就是节点的缓存空间无限大，但实际场景中，节点空间往往有限，尤其是在无线网络中，即使节点的缓存空间比较大，中继节点也只想分享少量的存储空间用于临时缓存。因此，如何管理节点缓存将会对路由协议产生重大影响，特别是间歇性连接和长延迟都要求数据长久的保存在节点中。

微观经济学中的边缘效用递减规律表明，当用户增加对某种商品的消费时，每消费一个单位的产品，虽然总的效用增加，但其所带来的效用的增加量是逐渐减小的，我们根据这一规律重点分析了一种基于缓存消息传播状态的缓存管理方案。所以，我们认为网络中对单一缓存产生过多的冗余副本是不好的，因为冗余副本会占据大量的缓存空间，当缓存溢出时，会限制其他信息获得被缓存的可能性。而我们所分析的缓存管理方案，根据缓存消息的传播状态，例如缓存消息在网络中总的副本数量和传播速度，来执行缓存管理方案。当某些节点的缓存空间满时，拥有副本较多且传播较快的缓存将会被优先丢弃，同样这些缓存被转发的优先级较低。该缓存管理方案解决了缓存间的公平性问题，因而改善了总体性能，仿真结果表明，同其他缓存管理方案相比，本文所分析的基于缓存传播状态的管理方案能够提高交付比率，减少开销比率。

4.2 系统模型与分析

4.2.1 系统模型

假设网络中有 N 个移动节点：1 个源节点，1 个目的节点和 N-2 个中继节点。源节点可以将消息直接传递给目的节点，也可以通过中继节点传递。

首先，我们来介绍本章接下来将要使用的基本定义和符号，以及状态的假设。

网络中的每种类型的节点都有同样大小的缓存空间，它们也拥有同样的传播范围。使用 $X_i(t)$ 和 $X_j(t)$ 分别表示在时刻 t 节点 i 和节点 j 的位置，γ 代表节点的传播范围。

定义 1 连接时间：两个节点连接的持续时间。在时刻 t_1，节点 i 和 j 在彼此的通信范围内，则两个节点的连接时间为 $\min_t\{(t-1)-t_1 : \| X_i(t)-X_j(t) \| > \gamma\}$。

定义 2 连接间隔时间：两个节点不在彼此通信范围内的时间。在时刻 t_1 两个节点进入了彼此的通信范围内，而在 t_2 时刻超出彼此的通信范围，t_2 的值为 $t_2 = \min_t\{t : \| X_i(t)-X_j(t) \| > \gamma\}$，则连接间隔时间 $\min_t\{t-t_2 : \| X_i(t)-X_j(t) \| \le \gamma\}$。

定义 3 丢弃间隔时间：节点丢弃消息或缓存副本的间隔时间。

相比于连接间隔时间而言，我们认为可以忽略消息的传输时间，并且假设每个消息都有生存时间（Time-to-Live，TTL）。当消息或者缓存副本的生存时间到达时，源端节点或者中继节点将会丢弃它们。一些研究表明，一些流行的的移动模型，如 Random Waypoint、Random Walk 和 Community-Based，它们的连接间隔时间服从指数分布或指尾分布[53][54]，为简单起见，要求连接间隔时间服从指数分布。我们假设丢弃间隔时间是独立同分布的随机变量，服从指数分布。

4.2.2 系统分析

我们通过一个单独的消息 m 来分析消息的传播，网络中携带消息 m 或其副本的节点称为被感染节点，其他节点称为未感染节点。在时刻 t，被感染的节点数为 x，未感染节点数为 s。假设网络中有 N 个节点，则有

$$s + x = N \tag{4.1}$$

根据假设，两个节点的接触率为 λ，则单位时间内感染节点将会遇见 $\lambda s/N$ 个未感染节点，那么单位时间内新获得感染节点数为 $\lambda sx/N$。同理，假设丢弃率为 μ，单位时间内节点从感染变成未感染的数量为 μx，则感染节点的增长率为

$$\frac{dx}{dt} = \lambda sx/N - \mu x \tag{4.2}$$

当节点与其他节点互联且缓存溢出时，节点会丢弃缓存数据，假设节点间的接触率大于丢弃概率，即 $\lambda > \mu$，我们定义

$$\rho = \lambda/\mu \tag{4.3}$$

将公示（4.1），（4.2）和（4.3）结合在一起得到

$$\frac{\mathrm{dx}}{dt} = -\lambda x[\frac{x}{N} - (1 - \frac{1}{\rho})] \tag{4.4}$$

被感染的节点数为

$$\mathrm{x} = N(1-1/\rho)\frac{1}{1+[N(1-1/\rho)-1]e^{-\lambda(1-1/\rho)t}} \tag{4.5}$$

我们可以从公式（4.3）和（4.5）中看出，x 的增长率取决于它的初始值，当 x 的初始值设为 1 时，在传播的初始阶段 x 增长较快，x 的极限值随着 ρ 的增加而增长：

$$\mathrm{x}(\infty) = N(1-1/\rho) \tag{4.6}$$

我们使用效用函数 $\mathrm{U}(x)$ 来表示每个消息或缓存副本被成功交付的可能性，它是关于 x 的递增函数，也就是说越多的节点缓存某个消息，则该消息越有可能被传递到目的节点，即

$$\frac{\mathrm{d}U(x)}{dx} \geq 0 \tag{4.7}$$

根据边际效用递减规律，当交付可能性达到一定程度时，网络中消息副本数量的增长对于交付可能性的帮助越来越低，即

$$\lim_{x \to N} \frac{dU(x)}{dx} = 0 \tag{4.8}$$

注意到，在缓存管理方案中，被感染的节点数目小于等于网络中总的节点数，因此，每个消息传递给目的节点的可能性也有一个上限，即

$$\lim_{x \to N} U(x) = 1 \tag{4.9}$$

通常假设效用函数 $\mathrm{U}(x)$ 具有一些规律属性，例如在某些分段上具有连续可微性，当这些属性运用到公式（4.7）、（4.8）和（4.9）时，我们可以得到当 x 大于某一个值时，$\mathrm{U}(x)$ 是凹函数，即

$$\exists \mathrm{c} : U(x) < 0, \forall x \geq c \tag{4.10}$$

因此，当网络中某个消息的缓存副本较少时，$\mathrm{U}(x)$ 是凸函数，从整个网络看，如果该消息被丢弃，则丢弃它而减少的效用高于增加另一个消息的缓存副本所产生的效用；而当一个消息在网络中缓存副本较多时，$\mathrm{U}(x)$ 具有凹函数性质，如果该消息被丢弃，则丢弃它减少的效用要低于增加另一个消息的缓存副本所产生的效用。

通过以上分析，我们可以得到，网络中某个消息具有更多数量的缓存副本，则该消息有更多的被传送的机会。这些消息更有可能到达目的节点，当我们控制这些消息的缓存副本数量的增长，虽然消息到达目的节点的概率会减小，但损失的效用会被限定在比较低的水平。对于一个单独消息来讲，限定消息缓存副本数量的增长可能会降低它们的交付比率，但同样的，给其他缓存副本较少的消息更多地被传送机会，增长的效用大于控制消息副本过多的消息而造成的减少效用，因此，会提高总体性能。

4.3 缓存替换与调度方案

4.3.1 缓存替换方案

由于网络的间歇性连接，一个节点不能够准确地知道一个特定消息的全局信息，当节点相遇时，可以利用统计学来评估消息的传播状态。我们介绍了两种度量指标来衡量消息的优先级，分别是消息副本的数量和消息的传播速度。消息的副本越少，则优先级越高；消息的传播速度越慢，优先级越高。

本章中，使用 R_m^i 表示节点 i 所知道的消息 m 的复制数量，明显可以看出，R_m^i 的值越大，消息 m 在网络中的扩散能力越强，同时也意味着网络中消息 m 的复制数量越多；反之，R_m^i 的值越小，消息 m 在网络中的副本就越少。节点 i 将优先丢弃 R_m^i 值较大的消息。

每个节点都有一个简单的矢量，矢量中的每个表项由四个部分组成，包括消息 ID，复制数量，初始 TTL 值和当前 TTL 值。使用简易矢量来更新消息的转发状态，当两个节点相遇时，彼此交换简易矢量。由于同消息相比，简易矢量所占空间非常小，因此不考虑简易矢量的额外开销。当一个节点接收到新消息时，节点在简易矢量中增加一个表项；当节点丢弃消息时，对应的简易矢量中的表项依然保持，直到消息的生存时间到达。当网络中产生一个新消息时，它的复制数量的初始值设为 1。

下面我们以一个消息为例，描述节点对消息副本数量的管理过程。

节点 i 携带有信息 m，节点 j 未携带消息 m，当两者相遇后，消息 m 的复制数的变化有以下两种情况：

（1）一种情况是节点 j 被选中为消息 m 的中继节点，如果节点 j 不包含消息 m 的任何信息，则两个节点中消息 m 的复制数都被设置为 R_m^i+1；否则，两个

节点交换简易矢量，并设置消息 m 的复制数为 $\max(R_m^i, R_m^j)+1$。

（2）另一种情况是，节点 j 未被选中为消息 m 的中继节点，若节点 j 不包含消息 m 的任何信息，则两个节点中消息 m 的复制数都被设置为 R_m^i；否则，两个节点交换简易矢量，并设置消息 m 的复制数为 $\max(R_m^i, R_m^j)$。

当不同的消息有着相同的估计复制数，我们使用另一种度量来衡量消息，参数 Rate 描述了一个消息的传播速度，定义如下：

$$\text{Rate} = \frac{K_m}{TTL_{init} - TTL} \tag{4.11}$$

K_m 表示缓存的消息从生成到现在所经历的跳数，我们的缓存管理方案把跳数同每个消息联系起来，传播速度越快的消息在网络中有着越多的消息副本。

缓存替换算法伪代码

```
If(m.size>Buffer.size)
    deleteMessage(m);
While(Buffer.freeSize<m.size) {
    msg = minPriority(messages in Buffer U m);
    If(msg == m)
        deleteMessage(msg);
    Else {
        deleteMessage(msg);
        Buffer.freeSize += msg.size;
    }
}
```

当节点的缓存空间有空余时，它就可以接收新的消息，否则就会根据上面提到的优先级，比较当前消息和节点缓存中的所有消息的优先级，低优先级的消息将会被丢弃。MTSBR（Message Transmission Status-Based Replacement）缓存替换算法伪代码如上所示。

4.3.2 缓存调度算法

当两个节点处于双方的通信范围内，由路由算法决定将哪些消息转发给下一个的中继节点，这些消息称为准备集（Ready Set，RS），节点会将准备集中所有消息转发给中继节点。但由于网络带宽受限或链路不稳定的原因，准备集中的

消息不一定都能成功地转发给中继节点，另外中继节点是否拥有足够的缓存空间来存储这些消息并不在路由算法的考虑范围之内，若是转发给中继节点的消息，因为节点缓存空间溢出而被丢弃，显然是对网络资源和节点开销的一种浪费，因而决定消息转发给中继节点的顺序非常重要。为了解决这个问题，我们采用基于消息传播状态的缓存调度算法（Message Transmission Status-Based Scheduling，MTSBS），MTSBS 算法的伪代码如下所示：

缓存调度算法伪代码

M_i：节点 i 中路由算法选中的准备集

$M_i(k)$：M_i 中第(k+1)个消息

If (M_i.occupancy < node j's freebuffer)

　　Sending　M_i　to node j;

If (the lowest priority in　M_i　> the highest priority in　M_j) {

　　For (k = 0 ; k <　M_i.size() ; k++) {

　　If (M.occupancy < node j's freebuffer)

　　M.add($M_i(k)$);　　}

　　Sending M to node j;

}

Sending ($\mathrm{Top}_{buffersize}(M_i + M_j) - M_j$) to node j.

首先，缓存调度算法 MTSBS 根据准备集中消息的优先级，将这些消息按递减顺序排列，消息副本数量较少和传播速度较慢的消息，其优先级就越高；消息副本较多且传播速度较快的消息，其优先级就越低，在任何情况下，拥有高优先级的消息将得到转发的机会。然后，MTSBS 选择哪些消息将转发给中继节点，若准备集中消息的最高优先级比中继节点中的缓存消息的最低优先级还低，那么中继节点中可用的缓存空间将用于存储转发消息；如果准备集中的消息的最低优先级比中继节点中的缓存消息的最大优先级还要高，那么节点将转发所有的消息给中继节点，若中继节点发生缓存溢出，则替换掉中继节点接收转发消息之前缓存的消息副本；其他情况下，合并准备集和中继节点中的所有消息，对合并后的所有消息进行优先级排序，按照中继节点的存储空间大小存储消息，最后将中继节点中没有的消息转发给中继节点。

4.4 仿真实验

4.4.1 仿真环境设置

我们使用 ONE 仿真模拟器[55]，采用基于洪泛式的 epidemic 路由协议[56]对缓存管理方案进行了模拟仿真，仿真中使用了如下性能指标：

交付比率：交付的消息占总消息数量的比例。

开销比率：成功交付一个消息所用的平均中继路由节点数量。

平均时延：目的节点接收到所有消息的平均时延。

我们在 ONE 平台上采用了一个默认的场景来仿真缓存管理方案，具体环境设置如下：六组节点分布在 4500m×3500m 大小的地方，分组 1 和分组 2 中的节点代表行人，这些节点的移动速度为 1.8~5.4km/h；分组 3 的节点表示汽车，移动速度为 10~50km/h。分组 1~3 中节点的缓存空间大小为 5~50MB，每个分组的节点数量为 20~50 个，分组的停滞时间为 0~120s。分组 4~6 的节点表示有轨电车，这些节点的移动速度为 25~36km/h，每组固定包含 2 个节点，缓存空间大小为 50MB，分组的停滞时间为 10~30s。每个节点的通信范围为 10m，节点之间的数据传输速度为 2Mbps。仿真时间设定为 12 小时，每 25 到 35 秒产生一个新消息，其生存周期为 6 小时，消息的大小为 0.5~1MB。

4.4.2 仿真结果

（1）缓存替换方案

实验中，我们将 MTSBR（Message Transmission Status-Based Replacement）与如下三种缓存替换方案进行了比较：

DO（Drop Oldest）：首先丢弃生存周期最小的缓存消息副本。

DF（Drop Front）：内容缓存器类似于一个栈，后进先出，首先删除栈顶的缓存消息副本。

HBD（History-Based Drop）：一种基于预估的关于缓存消息的全局信息的分布式消息丢弃方案，旨在最优化某个性能指标[57]，本实验中的性能指标是交付比率。

图 4.1，4.2，4.3 反映了缓存大小对路由性能的影响，该场景下，行人和汽车分组中，每个分组有 40 个节点，节点缓存大小为 5~50MB，所有的有轨电车分组中节点缓存大小为 50MB。

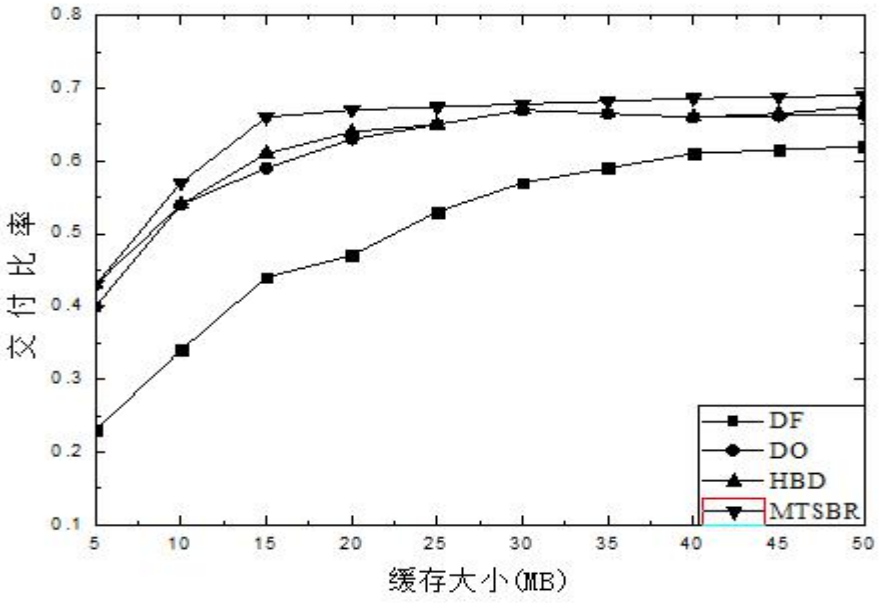

图 4.1　各缓存策略在不同缓存大小下的交付比率

图 4.1 表明，随着缓存空间的增加，缓存策略的交付比率也随之增大，但是当缓存大小增加到一定程度时，DO 和 HBD 的交付比率反而有所下降，虽然 HBD 和 DO 考虑了消息副本的数量，但它们都不关心消息的传播能力。DO 仅仅考虑局部消息副本的数量而不考虑整个网络中消息的状态。当某一消息在网络中达到一定数量时，在缓存满时，MTSBR 方案虽然丢弃网络中消息副本数量较多的消息，但剩余的消息副本仍然能够将消息扩散至目的节点，同时新产生的消息获得了更多地传输可能性，所以 MTSBR 的交付比率较高。

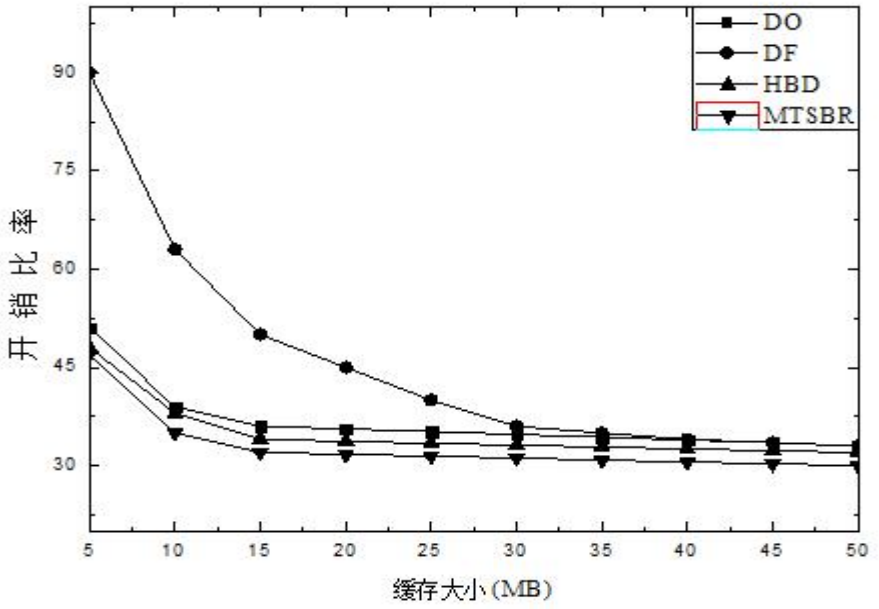

图 4.2　各缓存策略在不同缓存大小下的开销比率

从图 4.2 可以看出，随着缓存空间的增加，开销比率逐渐减小，这是因为，缓存空间增大，节点可以缓存更多地消息副本，减少了消息副本的替换，大大增大了副本在整个网络中的数量，因而可以更快的满足目的节点的请求，减少中继节点的跳数。从图 4.2 还可以看出，DF 策略的开销性能最差，因为一些新消息在传播扩散开始阶段就被丢弃，这使得被丢弃的新消息需要进行重传，导致网络开销较大。而 MTSBR 方案可以限制哪些消息副本较多且传播较快的消息的进一步大量扩散，将更多地传输机会留给新产生的消息，避免新消息在一些中继节点中在初始阶段被丢弃而重传，因而开销比较小。

图 4.3 表明，缓存空间增大，消息的平均时延降低，相比于其他缓存替换策略，MTSBR 方案可以减少消息副本较多且传播较快的消息的进一步大量扩散，虽然这有可能会替换掉那些快要达到目的节点的消息，但由于这些消息本身在网络中副本较多且传递较快，因而不大影响这些消息的扩散性，给新消息更多地传播扩散机会，能够在整体上使得所有消息的平均时延有效下降。

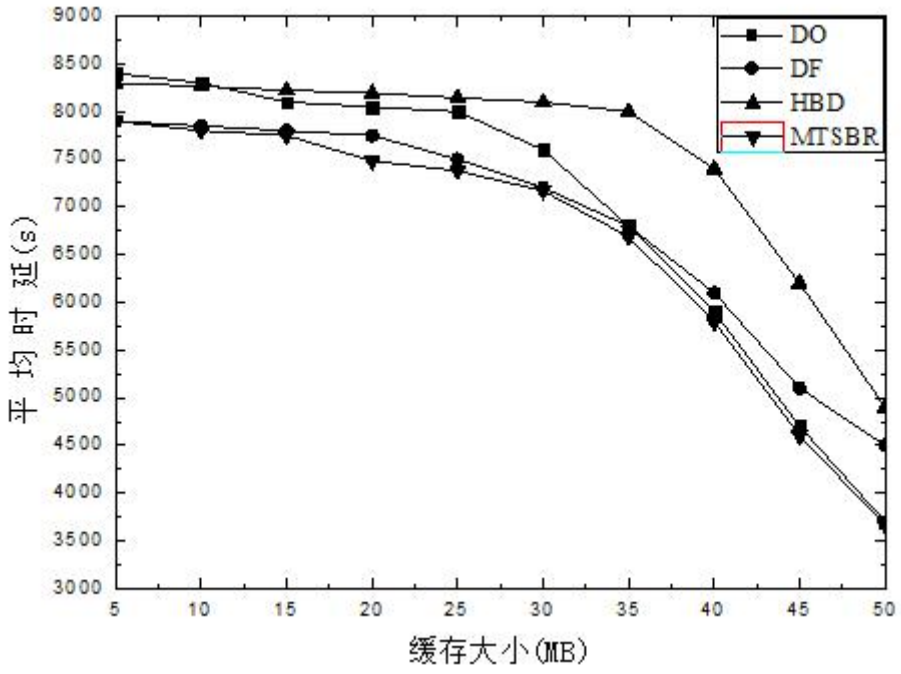

图 4.3　各缓存策略在不同缓存大小下的平均时延

图 4.4，4.5，4.6 反映了行人和骑车分组中增长的节点数量对性能的影响，有轨电车中的节点数固定为 2。

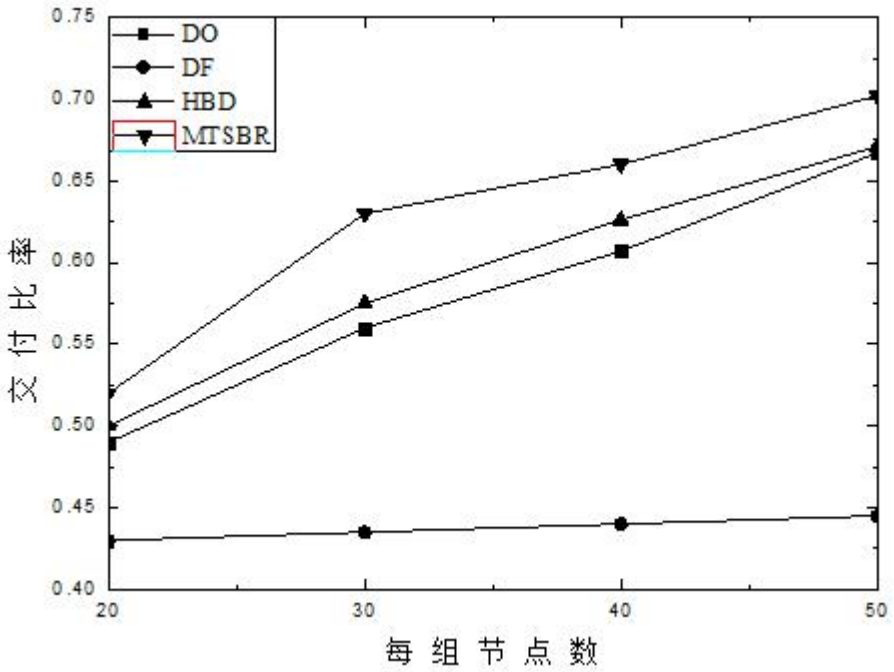

图 4.4　各缓存策略在不同节点数量下的交付比率

从图 4.4 中可以看出，随着节点数量的增加，交付比率逐渐提高，但 DF 的交付比率几乎没有太大的变化，这是因为 DF 在决定丢弃哪些消息时仅仅考虑局部缓存的状态，而其他策略则考虑整个网络的消息。当网络规模增大时，HBD 评估网络中消息的全局信息变得更加困难，而这对于 MTSBR 的影响要小，因此 MTSBR 的交付比率比 HBD 高。

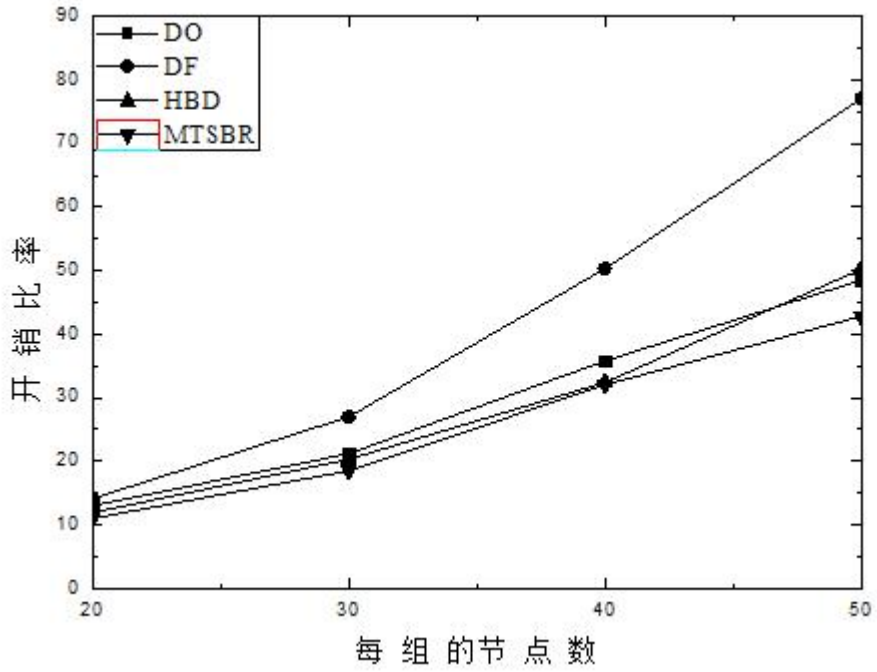

图 4.5　各缓存策略在不同节点数量下的开销比率

图 4.5 显示，随着网络规模的增大，即节点数量的增多，网络开销也相应的增大，消息被传递到更多的节点中，并被进一步复制，因此各种策略的开销比率不降反升。当节点数量增多时，带来了更多地传输机会，缓存竞争变得越加激烈。MTSBR 限制了有较多消息副本的消息的进一步扩散，因而网络开销比率增长最少。

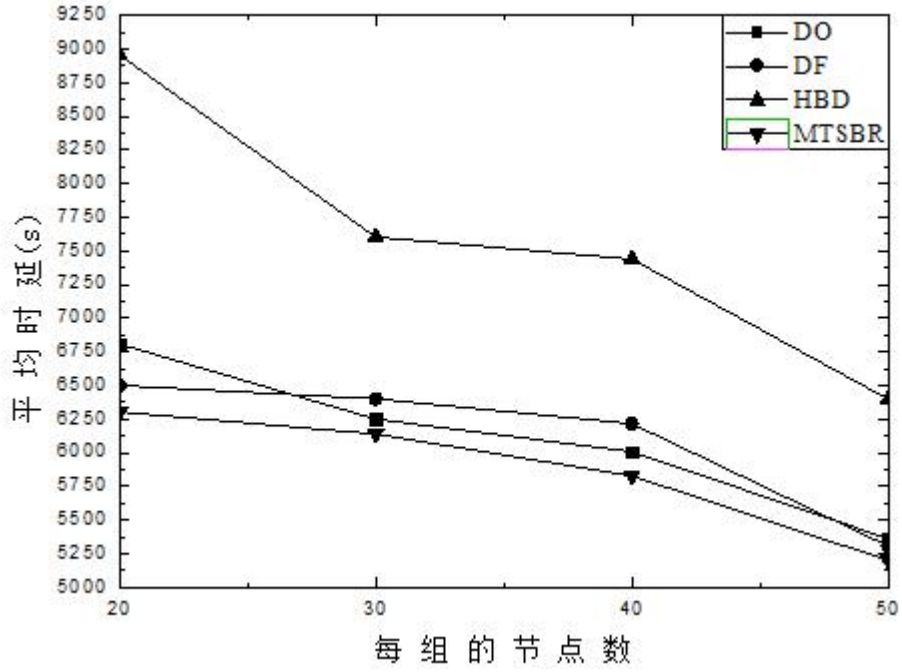

图 4.6　各缓存策略在不同节点数量下的平均时延

图 4.6 表明，随着节点数量的增加，消息到达的平均时延降低，这是因为有更多地节点能够中继转发消息，消息的传输机会增加。HBD 对快要到达目的节点的消息丢弃的可能性较大，而致使平均时延最高，MTSBR 虽然增加了消息副本较多且传播较快的消息的丢弃可能，但这些消息的扩散性较高，到达目的节点所受影响不大，且为新消息提供了更高的传输机会，因而能降低平均延时。

（2）缓存调度方案

实验中，我们将 MTSBS（Message Transmission Status-Based Scheduling）与如下两种调度方案进行比较：

FIFO：当两个节点相遇时，最近收到的消息将在最后转发。

Random：随机转发缓存中的一个消息。

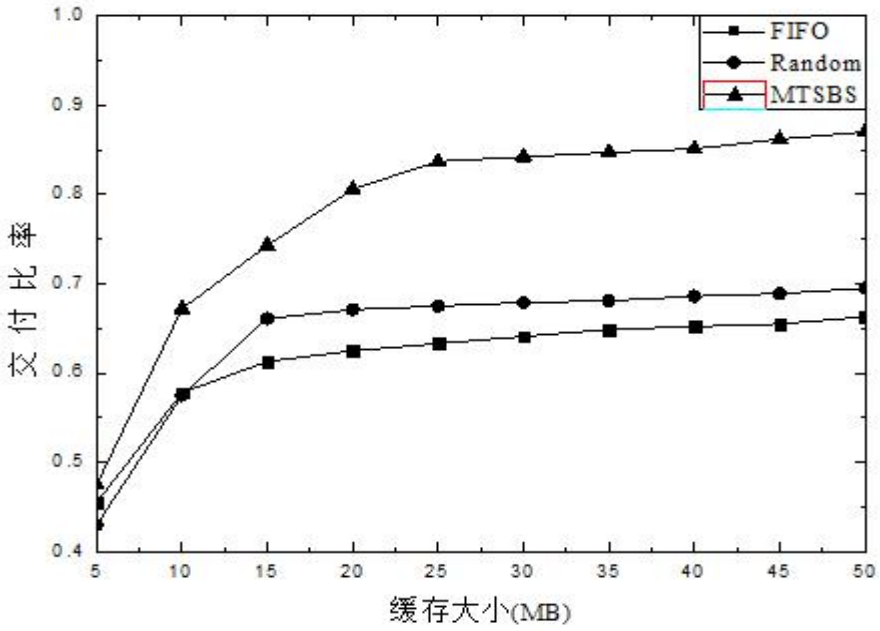

图 4.7　各缓存策略在不同缓存大小下的交付比率

图 4.7 展示了 MTSBS、FIFO 和 Random 的交付比率的对比，Random 策略在一定程度上能够为消息提供公平性，所以 Random 的交付比率比 FIFO 要高，而 MTSBS 是基于效用值传播消息，因此交付比率性能显著提高。从图 4.8 可以看出，MTSBS 的开销比率最低，这是因为调度方案将决定传输哪些消息。调度机制考虑缓存限制，不会传递那些在下一跳中继节点就会被丢弃的消息。图 4.9 显示，相较于 FIFO 和 Random 策略，MTSBS 的平均时延最低。

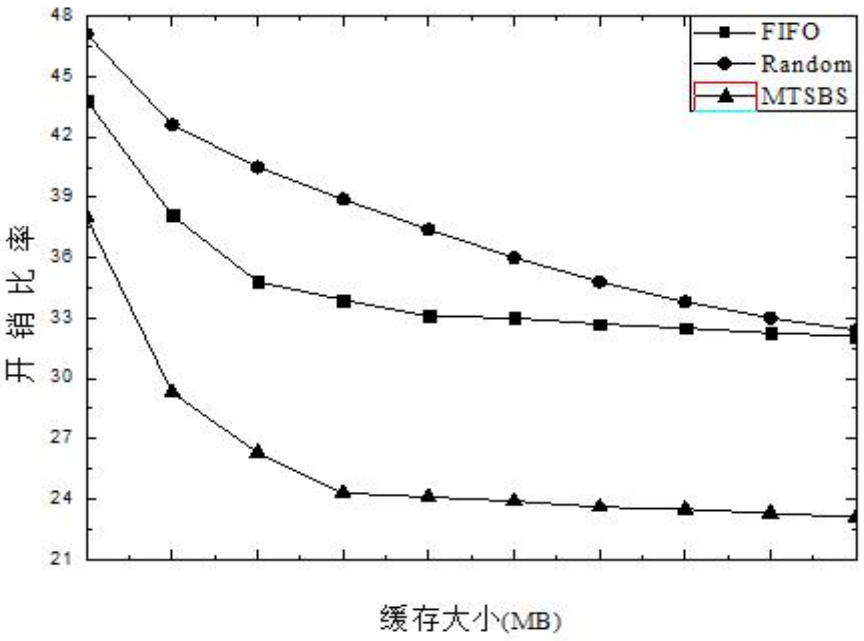

图 4.8　各缓存策略在不同缓存大小下的开销比率

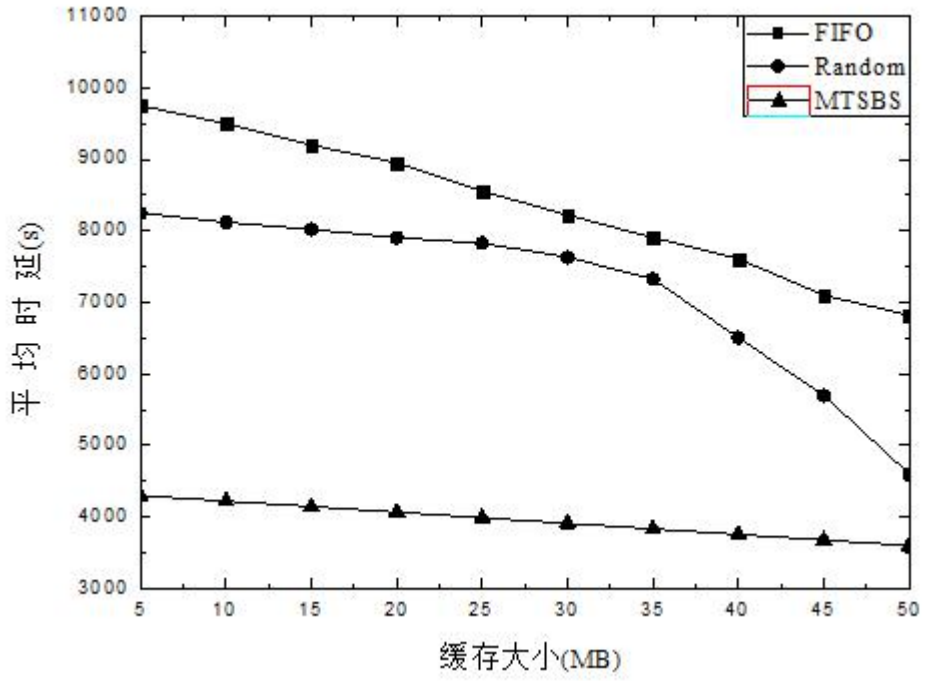

图 4.9　各缓存策略在不同缓存大小下的平均时延

4.5 本章小结

在一些延迟容忍网络环境中，节点的缓存空间和能量受限，低效率的缓存管理方案可能会对路由算法产生不良影响，造成网络拥塞，导致整体网络性能表现不佳。本文分析的基于消息传播状态的节点缓存管理方法，该方法能够抑制网络中消息副本较多且传播速度较快的消息的进一步扩散，使副本数量较少、传播速度较慢的消息得到更多地传输机会。仿真结果显示，相较于其他常见缓存替换和调度策略，基于消息传播状态的缓存管理方案能够提高路由算法的交付比率，且开销比率和平均时延均优于常见缓存替换和调度策略。

第五章 总结与展望

本文研究了两种未来网络体系架构：信息中心网络和延迟容忍网络，总结了两种网络架构下国内外存储机制的研究现状，分析了两种体系架构下的常用存储机制，在人们对网络需求快速增加的背景下，未来网络缓存存储机制是为了解决现有互联网网络冗余大、资源利用率低的弊端。

未来网络更多地是以内容为中心，用户只关心所需内容本身，而不关心内容的存储位置，通过对内容名字在全局网络中唯一标识，这样就可以通过内容名字对内容进行定位、路由和传输，网络中的路由节点都具有一定的缓存能力。本文主要选取了两种网络体系作为研究代表，即信息中心网络 ICN 和延迟容忍网络 DTN。

ICN 网络下，目前大多数对缓存存储的研究集中在缓存放置策略和缓存替换策略，本文研究了一种数据块缓存位置与搜索方案 CLS，CLS 的基本思想是：在中间路由节点中添加一个缓存记录，沿着请求端到服务端的路径将缓存内容存储在命中节点的下游节点，同时删除本地节点中的缓存，保证路径上有且只有一份相同的缓存数据；而当节点缓存满时，则将缓存返回至上游节点，删除本地缓存。进行搜索时，路径上搜索简单，而对于路径外搜索，则根据缓存记录与事先设定的阈值，决定将请求向上游节点转发还是向下游节点转发。我们使用 ccnSim 仿真平台，在缓存命中率、缓存命中距离、平均下载时延和带宽消耗这些性能指标上，结合 LRU 替换策略，将 CLS 与常见缓存放置策略进行了对比，结果表明 CLS 优于常见缓存放置策略。未来我们将考虑不同的网络拓扑结构和一些条件受限的情况，例如缓存空间大小受限、对服务质量要求高的视频服务等，在仿真实验上，还可以考虑新的性能指标，比如路由节点缓存内容存储与替换的开销等。此外，ICN 缓存对象是 chunk，此前分析通常认为 chunk 大小固定，但在实际情况中，比如视频等大数据，chunk 之间可能具有关联性，如何利用缓存数据块的相关性也是下一步的研究方向。

DTN 网络中，本文研究了一种基于消息传播状态的缓存替换与调度方案，该方案的基本思想是：根据微观经济学边际效应递减规律，优先替换网络中消息副本较多且传播扩散速度较快的消息缓存，在消息调度上，则优先转发消息副本较少且传播扩散速度较慢的消息。消息副本较多且传播较快的消息，其本身扩散性

强，高优先级被替换、低优先级被转发，对于其传递到目的节点的影响较小，为其他消息获得更多地传输机会提高了可能性，总体性能得到提升。我们在 ONE 平台上，基于洪泛式路由算法 Epidemic，在交付比率、开销比率和平均时延这些性能指标上，对比了本方案同常见缓存替换方案和调度方案，仿真结果表明，本文研究的缓存管理方案优于其他常见缓存管理方案。未来我们将会考虑添加新的性能指标对比，比如网络负载和缓存消息存储与替换的成本等，同时在更多的网络拓扑下分析该缓存存储方案的性能表现。

参考文献

[1] Cisco Visual Networking Index:Forecast and Methodology, 2014–2019.[OL]，2015.
[2] Nelson T. Literary machines: the report on, and of, project Xanadu concerning word processing, electronic publishing,hypertext, thinkertoys, tomorrow' s intellectual revolution,and certain other topics including knowledge, education and freedom[M]. Sausalito, California: Mindful Press, 1981.
[3] Baccala B. Data-oriented networking, Internet draft[EB/OL] (2002)[2013-01]. http://tools.ietf.org/html/draft-baccala-data-networking-00.
[4] M.Gritter and D.R.Cheriton. (2000, Jul.) Triad: A new next-generation architecture. [Online]. Available: http://www-dsg.stanford.edu/triad/mternet.
[5] Koponen T, Chawla M, Chun B G, et al. A data-oriented (and beyond) network architecture[J]. Acm Sigcomm Computer Communication Review, 2010, 37(4):181-192.
[6] Dannewitz C, Pentikousis K, Rembarz R, et al. Scenarios and research issues for a network of information[C]//Proceedings of the 4th International ICST Mobile Multimedia Communi-canons Conference (MobiMedia' 08), Oulu, Finland, July 7-9, 2008.
[7] Jacobson V, Smetters D K, Thornton J D, et al. Networking named content[C]// Proceedings of the 5th International Con-ference on Emerging Networking Experiments and Technologies (CoNEXT ' 09). New York, NY, USA: ACM, 2009: 1一12.
[8] United State NSF Future Internet Architecture project, http://www.nets-fia.net/[OL], 2011-12-12.
[9] NDN project[EB/OL]. http://anr-connect.org/, 2011-12-12.
[10] MobilityFirst project[EB/OL].http://mobilityfirst.winlab.rutgers.edu/,2011-12-12.
[11] NEBULA project[EB/OL]. http://nebula.cis.upenn.edu/, 2011-12-12
[12] XIA project[EB/OL]. http://www.cs.cmu.edu/~xia/, 2011-12-12.
[13] Dall'Asta M, Derlindati E, Ardigò D, et al. Design considerations for a network of information.[C]// Proceedings of the 2008 ACM Conference on Emerging Network Experiment and Technology, CoNEXT 2008, Madrid, Spain, December 9-12, 2008. 2008:387–392.
[14] Jokela P, Zahemszky A, Rothenberg C E, et al. LIPSIN: line speed publish/subscribe inter-networking. Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication, New York, NY, USA, 2009: 195~206
[15] Jacobson V, Smelters D K, Thornton J D, et al. Networking Named Content. CoNEXT[J]. Conext, 2009, 55(1):1--12.
[16] Wang J M, Zhang J, Bensaou B. Intra-AS cooperative caching for content-centric networks[C]// ACM SIGCOMM Workshop on Information-Centric NETWORKING. 2013:61-66.
[17] Laoutaris N, Che H, Stavrakakis I. The LCD interconnection of LRU caches and its analysis ☆[J]. Performance Evaluation, 2006, 63(7):609-634.
[18] Detlefsen M, Luker M. WAVE: Popularity-based and collaborative in-network caching for content-oriented networks[C]// Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on. IEEE, 2012:316-321.
[19] Wei Koong Chai, Diliang He, Ioannis Psaras, et al. Cache “less for more” in information-centric networks (extended version)[J]. Computer Communications, 2013, 36(7):758-770.
[20] Psaras I, Chai W K, Pavlou G. Probabilistic in-network caching for information-centric networks[C]// Edition of the Icn Workshop on Information-Centric NETWORKING. ACM, 2012:55-60.
[21] Ming Z, Xu M, Wang D. Age-based cooperative caching in Information-Centric Networks[C]// IEEE Conference on Computer Communications Workshops. 2012:268 - 273.
[22] Zhang B X, Neglia G, Kurose J, et al. Performance Modeling of Epidemic Routing,”Computer[C]// Networks. 2010.
[23] Krifa A, Barakat C, Spyropoulos T. An optimal joint scheduling and drop policy for Delay Tolerant Networks[C]// World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a. IEEE, 2008:1-6.
[24] Lindgren A, Phanse K S. Evaluation of Queueing Policies and Forwarding Strategies for

Routing in Intermittently Connected Networks[C]// International Conference on Communication System Software and Middleware, 2006. Comsware. 2006:1 - 10.
[25] Khelil A. Contact-based buffering for delay-tolerant ad hoc broadcasting[J]. Computer Communications, 2007, 30(16):3144-3153.
[26] Erramilli V, Crovella M. Forwarding in opportunistic networks with resource constraints.[C]// The Workshop on Challenged Networks, Chants 2008, San Francisco, California, Usa, September. 2008:41-48.
[27] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in Proc. 2009 ACM CoNEXT Conf., Rome, Italy, 2009, pp. 1–12.
[28] Psaras I, Wei K C, Pavlou G. In-Network Cache Management and Resource Allocation for Information-Centric Networks[J]. IEEE Transactions on Parallel & Distributed Systems, 2014, 25(11):2920-2931.
[29] Laoutaris N, Che H, Stavrakakis I. The LCD interconnection of LRU caches and its analysis [J]. Performance Evaluation, 2006, 63(7):609-634.
[30] Li Y, Lin T, Tang H, et al. A chunk caching location and searching scheme in Content Centric Networking[C]// Communications (ICC), 2012 IEEE International Conference on. IEEE, 2012:2655 - 2659.
[31] Ho C Y, Ho C Y, Tseng C C. A case study of cache performance in ICN — Various combinations of transmission behavior and cache replacement mechanism[C]// International Conference on Advanced Communication Technology. IEEE, 2015.
[32] Petev P G, Wintergerst M. Least frequently used eviction implementation: US, US 7552284 B2[P]. 2009.
[33] Lee D, Choi J, Choe H, et al. Implementation and Performance Evaluation of the LRFU Replacement Policy[C]// euromicro. IEEE Computer Society, 1997:0106-0106.
[34] Delay tolerant networking research group[EB/OL]. 2012-02-23.http://www. dtnrg. org/.
[35] InterPlaNetary Internet Project[EB/OL].http://www.ipnsig.org/.
[36] Fall K. A delay-tolerant network architecture for challenged internets[C]// Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. ACM, 2003:27-34.
[37] Psaras I, Clegg R G, Landa R, et al. Modelling and Evaluation of CCN-Caching Trees[M]// NETWORKING 2011. Springer Berlin Heidelberg, 2011:78-91.
[38] Carofiglio G, Gallo M, Muscariello L, et al. Modeling data transfer in content-centric networking[C]// Teletraffic Congress (ITC), 2011 23rd International. IEEE, 2011:111 - 118.
[39] Muscariello L, Carofiglio G, Gallo M. Bandwidth and storage sharing performance in information centric networking[C]// ACM SIGCOMM Workshop on Information-Centric NETWORKING. ACM, 2011:26-31.
[40] Carofiglio G, Gehlen V, Perino D. Experimental Evaluation of Memory Management in Content-Centric Networking[J]. IEEE International Conference on Communications, 2011, 41(4):1-6.
[41] Che H, Wang Z, Tung Y. Analysis and design of hierarchical Web caching systems[J]. Proceedings - IEEE INFOCOM, 2001, 3:1416 - 1424.
[42] Tang X, Chanson S T. Coordinated En-Route Web Caching[J]. Computers IEEE Transactions on, 2002, 51(6):595-607.
[43] Borst S, Gupta V, Walid A. Distributed caching algorithms for content distribution networks[C]// Conference on Information Communications. IEEE Press, 2010:1478-1486.
[44] Laoutaris N, Che H, Stavrakakis I. The LCD interconnection of LRU caches and its analysis ☆[J]. Performance Evaluation, 2006, 63(7):609-634.
[45] Laoutaris N, Syntila S, Stavrakakis I. Meta algorithms for hierarchical Web caches[C]// Performance, Computing, and Communications, 2004 IEEE International Conference on. 2004:445-452.
[46] Rosensweig E J, Kurose J. Breadcrumbs: efficient, best-effort content location in cache networks[J]. Proceedings - IEEE INFOCOM, 2009:2631 - 2635.
[47] ccnSim homepage[OL]. http://www.telecom-paristech.fr/ccnSim.
[48] L Breslao, P Cao, L Fan, S.Shenker, et al. "Web Caching and Zipf-LikeDistributions: EvidenceandImplications[J]. In Proc:of Infocomm'99, 1999, 1:126 - 134.

[49] Chlebus E, Brazier J. Nonstationary Poisson modeling of web browsing session arrivals[J]. Information Processing Letters, 2007, 102(5):187-190.
[50] Juang P, Oki H, Wang Y, et al. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet[C]// International Conference on Architectural Support for Programming Languages and Operating Systems. 2002:96-107.
[51] Hui P, Chaintreau A, Scott J, et al. Pocket switched networks and human mobility in conference environments[C]// Acm Sigcomm Workshop on Delay-tolerant Networking. ACM, 2005:244--251.
[52] Disruption tolerant networking[OL], http://www.darpa.mil/sto/programs/dtn/.
[53] Spyropoulos T. Performance analysis of mobility-assisted routing[C]// ACM MOBIHOC. 2006:49--60.
[54] Groenevelt R, Nain P, Koole G. Message delay in MANET[J]. Acm Sigmetrics Performance Evaluation Review, 2005, 33(1):412-413.
[55] Ker, Nen A, Ott, J, et al. The ONE simulator for DTN protocol evaluation[C]// Proceedings of the 2nd International Conference on Simulation Tools and Techniques for Communications, Networks and Systems, SimuTools 2009, Rome, Italy, March 2-6, 2009. 2009.
[56] Vahdat A, Becker D. Epidemic Routing for Partially-Connected Ad Hoc Networks[J]. Master Thesis, 2000.
[57] Krifa A, Barakat C, Spyropoulos T. Optimal Buffer Management Policies for Delay Tolerant Networks[C]// Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on. 2008:260 - 268.
[58] Zhang T, Chen W, Han Z, et al. Coalitional game theoretic approach for cooperative transmission in vehicular networks[C]// IEEE International Conference on Communications. IEEE, 2013:6179-6183.
[59] Sidi HBA, Chahin W, Elazouzi R, et al. Incentive Mechanisms based on Minority Game in Heterogeneous DTNs[J]. Azouzi, 2012.
[60] Alahdal T, Alsaqour R, Abdelhaq M, et al. Reliable Buffering Management Algorithm Support for Multicast Protocol in Mobile Ad-hoc Networks[J]. Journal of Communications, 2013, 8(2):136-150.
[61] Deng X, Chang L. Sociality-Based Buffer Management for Multicast in DTNs[C]// Fourth International Conference on Emerging Intelligent Data and Web Technologies. IEEE, 2013:508-514.
[62] Ramiro V, Dang D K, Baudic G, et al. A Markov chain model for drop ratio on one-packet buffers DTNs[C]// World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015 IEEE 16th International Symposium on a. IEEE, 2015:1-6.
[63] Liu F, Bai X. Research on the Buffer Management Algorithm in DTN[C]// Information Science and Control Engineering (ICISCE), 2015 2nd International Conference on. IEEE, 2015:442-446.
[64] Zhang M, Luo H, Zhang H. A Survey of Caching Mechanisms in Information-Centric Networking[J]. IEEE Communications Surveys & Tutorials, 2015, 17(3):1-1.
[65] Jiang X, Jun B I, Nan G, et al. A Survey on Information-Centric Networking:Rationales, Designs and Debates[J]. Wireless Communication Over Zigbee for Automotive I-nclination Measurement China Communications, 2015, 12(7):1-12.
[66] Abdallah E G, Zulkernine M, Hassanein H S. Detection and Prevention of Malicious Requests in ICN Routing and Caching[C]// IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. IEEE,2015.

致谢

时间像缓缓流淌的溪水，在不经意间悄然逝去，随着论文的完结，三年的研究生生活即将结束，在这即将毕业的时刻，我要衷心的感谢那些关心和帮助过我的老师、同学和朋友亲人。

首先感谢我的导师杨龙祥教授，从本科毕业设计到这三年研究生的时间里，杨老师一直悉心指导我进行科学研究，耐心地帮助我认识、理解研究课题的内容含义。他常常教导我们说，不要把毕业设计仅仅当成一项任务，而是要讲其当做今后学习和工作的一个重要的培养过程，在这个过程中，我们应该学会独立自主的学习，能够静下心来认真研究，掌握科学研究的每个步骤，学会与人真诚的交流沟通。杨老师的独自自主培养方式、严谨的治学态度以及和蔼可亲的师者风范，令我在这个过程中受益匪浅，更会在今后的学习工作中受用终生。在此，向杨老师致以崇高的敬意和衷心的感谢。

我要衷心感谢那些曾经关心和指导过我的老师以及那些陪伴我度过三年研究生生活的同学们，尤其是我的同门任美翠、卜旭阳、蒋燕燕、郑雪纯、胡悦、徐浩，我们相互学习、共同进步，还有我的室友武聪、靳丰远、冯盾、李迪阳、邵猛，给我以生活上的关心和照顾，感谢你们一路上的陪伴与帮助，令我的研究生生活倍感充实。

衷心的感谢我的爸爸妈妈，养育我二十五年，在这么多年的求学生涯中，是你们的无私奉献和默默关心，让我在遇到的大大小小的困难挫折时，能够不断地坚定向前，你们永远是我最坚强的后盾，即将踏入工作岗位，我会继续努力奋斗，踏实进取，用自己的辛勤劳动来回报你们的养育之恩。

谨向在百忙之中抽出时间审阅论文和参加答辩的各位老师表示衷心的感谢，是你们的指导，让我认识到自己的不足。最后，再次衷心地感谢我的导师杨龙祥教授以及帮助过我的老师、同学、朋友和家人！