

中文图书分类号: TP393
密 级: 公开
UDC: 004
学 校 代 码: 10005



硕 士 专 业 学 位 论 文

PROFESSIONAL MASTER DISSERTATION

论 文 题 目: 命名数据网络拥塞控制策略的研究

论 文 作 者: 刘晓雨

专业类别/领 域: 软件工程

指 导 教 师: 张丽 副教授

论文提交日期: 2020 年 6 月

UDC: 004
中文图书分类号: TP393

学校代码: 10005
学 号: S201761825
密 级: 公开

北京工业大学硕士专业学位论文

(非全日制)

题 目: 命名数据网络拥塞控制策略的研究

英文题目: RESEARCH ON CONGESTION CONTROL
STRATEGY OF NAMED DATA NETWORK

论 文 作 者: 刘晓雨

专业类别/领域: 软件工程

研 究 方 向: 数字媒体技术与应用

申 请 学 位: 工程硕士专业学位

指 导 教 师: 张丽副教授

所 在 单 位: 信息学部

答 辩 日 期: 2020 年 5 月

授予学位单位: 北京工业大学

独 创 性 声 明

本人声明所呈交的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，也不包含为获得北京工业大学或其它教育
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。

签 名： 刘晓雨

日 期：2020 年 06 月 06 日

关于论文使用授权的说明

本人完全了解北京工业大学有关保留、使用学位论文的规定，即：学校有
权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部
或部分内容，可以采用影印、缩印或其他复制手段保存论文。

（保密的论文在解密后应遵守此规定）

签 名： 刘晓雨

日 期：2020 年 06 月 06 日

导师签名： 张丽

日 期：2020 年 06 月 06 日

摘要

近年来,随着互联网的高速发展,人们对网络中数据的传输效率提出了更高的要求,当前互联网体系结构中存在的问题与弊端日渐突出,以往的 TCP/IP 体系架构遵循的端到端通信模式出现了很多弊端。随着科技技术的不断发展,越来越多新型网络架构涌现出来,在学术界掀起了一股研究热潮,命名数据网络(Named Data Networking, NDN)作为信息中心网络的代表,出现在人们的视野。目前关于命名数据网络的架构设计已日渐完备,而对于互联网体系架构普遍存在的关键问题的研究则尚待完善。网络拥塞作为网络体系架构设计的关键,同时也是网络能否高效通信的关键。网络拥塞本质上是网络资源供需关系的协调问题,在数据传输过程中,当网络中传输的内容超过网络能够容纳的上限时,供需关系失衡,就会发生网络拥塞。如何有效缓解网络拥塞,使数据正常传输是目前互联网发展中亟待解决的重点问题。目前 NDN 采取的拥塞控制方案主要分为两种:基于用户端的拥塞控制和逐跳的拥塞控制机制。前者采用的是 TCP/IP 体系架构中的超时机制,当超过一定时间后,请求还未得到满足,就判定出现了丢包的情况,并在用户端采用降低发送兴趣包速率的方式来缓解网络拥塞。后者是在路由转发的过程中进行逐跳的拥塞控制,通过路由节点调整本地的请求发送速率,进而恢复网络状态。这两种拥塞控制机制都存在一些弊端,首先,在拥塞检测方面,仍沿用 IP 网络中的超时重传机制,只有发生超时后,才会采取降低发送兴趣包速率的措施,让拥塞控制变得被动;其次在用户端采用 IP 网络中基于滑动窗口的速率调节方案,虽然能在一定程度上缓解网络拥塞,但在网络处于空闲状态时,并没有相关措施指出怎样利用这部分空闲带宽;最后逐跳的拥塞控制方案在中间节点调整兴趣包发送速率,并没有在路由转发层面考虑缓解网络拥塞的方案。

本文分别在用户端和逐跳式两个方面进行拥塞控制,针对前面提到的拥塞控制机制的弊端,提出了综合型拥塞控制策略,该策略主要利用早发现早预防的思想构造网络拥塞检测机制,实时检测网络流量,并做出相应反馈。在获知网络状态的基础上,分别对用户端采取调整网络请求发送速率的措施和中间路由节点采取基于路由接口排名的转发策略进行调节,从而实现网络性能的优化。本文的研究目的是在实际应用环境中,使命名数据网络能够发挥其数据传输方面的优势,避免由于网络拥塞的发生而造成性能方面的下降,从而为命名数据网络的实际应用贡献一份力量。

本文在 ndnSIM 上进行了拥塞控制策略的仿真模拟实验,将本文提出的综合型拥塞控制策略和 NDN 目前采用的拥塞调节策略进行了对比,发现综合型拥塞

控制策略在链路带宽利用率、路由节点内容命中率、用户总命中率方面占有一定优势，从而达到提高网络内容命中率，缓解网络拥堵的目的。

关键词：命名数据网络；拥塞控制；速率调节；转发策略

Abstract

In recent years, with the rapid development of the Internet, people have put forward higher requirements for the efficiency of data transmission in the network. The problems and drawbacks in the current Internet architecture have become increasingly prominent. The previous TCP / IP architecture follows the end-to-end There are many drawbacks in the communication mode. With the continuous development of technology, more and more new network architectures have emerged, and there has been a wave of research in the academic world. Named Data Networking (NDN), as a representative of the information center network, has appeared in people's vision. . At present, the design of the architecture of the named data network is becoming more and more complete, and the research on the key problems that are common in the Internet architecture is still to be improved. Network congestion is the key to the design of network architecture and the key to whether the network can communicate efficiently. Network congestion is essentially a coordination problem of the supply and demand of network resources. In the process of data transmission, when the content transmitted in the network exceeds the upper limit that the network can accommodate, the supply and demand relationship is unbalanced, and network congestion will occur. How to effectively alleviate network congestion and enable normal data transmission is a key issue to be solved urgently in the development of the Internet. At present, the congestion control scheme adopted by NDN is mainly divided into two types: user-based congestion control and hop-by-hop congestion control mechanism. The former uses the timeout mechanism in the TCP / IP system architecture. When the request is not satisfied after a certain period of time, it is determined that a packet loss has occurred, and the rate of sending interest packets is reduced at the user end to ease the network congestion. The latter is hop-by-hop congestion control during the route forwarding process, and the routing node adjusts the local request sending rate to restore the network state. Both of these congestion control mechanisms have some disadvantages. First of all, in terms of congestion detection, the timeout retransmission mechanism in the IP network is still used. Only after a timeout occurs, measures to reduce the rate of sending interest packets will be taken to make congestion control become Passive; secondly, a sliding window-based rate adjustment scheme in the IP network is adopted on the user side. Although it can alleviate network congestion to a certain extent, when the network is idle, there are no relevant measures to indicate how to use this part of idle bandwidth; The hop congestion control scheme adjusts the rate of interest packet transmission at the intermediate node, and does not consider a scheme to alleviate network congestion at the route forwarding level.

This paper discusses the optimization of network congestion status in two aspects, user-side and hop-by-hop. According to the characteristics of the problem under consideration, this paper mainly uses the idea of early detection and early prevention to construct a network congestion detection mechanism, detect network traffic in real time, and make corresponding feedback, and propose a comprehensive congestion control strategy. On the basis of knowing the state of the network, at the user end, the forwarding strategy based on the ranking of the routing interface is adopted to adjust the network request sending rate and the intermediate routing nodes to adjust, so as to achieve the optimization of network performance. The research purpose of this article is that in the actual application environment, the Named data network can take advantage of its data transmission and avoid performance degradation due to network congestion, thereby contributing to the practical application of the Named data network.

In this paper, a simulation experiment of congestion control strategy was carried out on ndnSIM. The comprehensive congestion control strategy proposed in this paper was compared with the congestion adjustment strategy currently adopted by NDN. The content hit rate and the total user hit rate have certain advantages, so as to achieve the purpose of improving the network content hit rate and alleviating network state congestion.

Keywords: NDN, Congestion control, Rate adjustment, Forwarding strateg

目 录

摘 要.....	I
Abstract.....	III
第 1 章 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.3 本文研究内容	4
1.4 本文结构	5
第 2 章 NDN 基础架构及相关研究工作	7
2.1 NDN 网络概述.....	7
2.1.1 NDN 沙漏架构.....	7
2.1.2 数据包类型和数据结构.....	8
2.1.3 命名规则.....	10
2.1.4 路由与转发.....	12
2.1.5 缓存机制.....	14
2.2 NDN 拥塞控制策略的相关研究	15
2.3 NDN 转发策略的相关研究.....	20
2.4 本章小结	23
第 3 章 综合型拥塞控制算法的拥塞检测设计	25
3.1 综合型拥塞控制算法设计思路	25
3.2 拥塞检测设计	27
3.2.1 拥塞检测设计核心思想.....	27
3.2.2 网络状态的划分	27
3.2.3 网络检测的结果反馈.....	29
3.3 本章小结	30
第 4 章 综合型拥塞控制算法的速率调节设计	31
4.1 基于接收端的速率控制	31
4.2 基于路由接口排名的转发策略	32
4.2.1 基于路由接口排名的转发策略的核心思想.....	32
4.2.2 数据结构分析.....	34
4.2.3 基于路由接口排名的转发策略具体流程.....	35

4.3 本章小结	37
第 5 章 算法的实验与分析	39
5.1 INDNSIM 模拟平台	39
5.2 模拟实验基本设计	41
5.3 算法的模拟实验	42
5.4 实验评估性能指标	44
5.5 实验结果	45
5.6 本章小结	51
结 论	53
参 考 文 献	55
攻读硕士学位期间所发表的学术论文	59
致 谢	61

第1章 绪论

1.1 研究背景及意义

随着互联网技术的高速发展,用户的不断增多,各类互联网应用的层出不穷,人们逐渐不满足于传统的互联网输出方式,传统的 TCP/IP 体系架构体现出其局限性,使得互联网的未来发展充满了挑战。端到端通信是 TCP/IP 网络的通信标志,协议规定地址是通信的核心,必须在地址已知的情况下才能实现用户的通信。虽然 TCP/IP 体系架构发展到今天已经取得了巨大的成就,框架体系日渐完善,但是科技愈加发达,“全民网民”的概念已不再是一个目标,随之而来的是呈指数增长的互联网用户,暴增的网络用户给 TCP/IP 体系架构协议带来了压力,传统的 IPV4 地址分配方式应对当前的用户规模已力不从心,因此 IPV6 协议和网络地址转换^[1](Network Address Translation,NAT)技术应用而生,解决当前用户规模下地址分配的问题。虽然 IPV6 能在一定程度上缓解 IPV4 的压力,但是这种延续已有思路再扩展的方式并不能从根本上解决用户暴增带来的问题。而且,随着互联网技术的不断发展,在网络中人们不仅仅共享资源,还逐渐有了获取资源的需求,内容信息的获取,取代了内容缓存节点的位置,成为人们关注的重点。这些对于 TCP/IP 架构来说都不再适合,信息中心网络^[2]的诞生,解决了传统网络架构的痛点,在信息中心网络中,学者们一致看好内容中心网络的发展前景,并投入了巨大的科研力量进行研究。通过不断的设计与改进,研究学者们提出了命名数据网络(Named Data Networking,NDN),NDN 也成为二十一世纪内容中心网络最具有代表性的项目。

命名数据网络中网络转发的核心信息是请求内容的名字,和 IP 网络相比,NDN 不再通过 IP 地址,而是通过带有层次结构内容的名称进行数据的传输,NDN 的转发过程依赖于名字前缀,而 IP 网络的转发过程则依赖于 IP 地址,在这一点上,NDN 和 IP 网络是不同的。命名数据网络中对提供内容的内容源来自何处并不限制,NDN 规定在请求分组中携带需要获取的内容的名字前缀,通过路由节点转发出去,进而收到请求的数据。虽然 NDN 和其他网络架构相比具有一定的优势,解决了现有网络架构普遍存在的地址不足问题,但由于 NDN 结构与传统体系架构的不同,NDN 网络存在一些其他的限制和痛点,需要进一步的研究与完善。

NDN 作为未来网络架构研究的重要项目,学者们投入了极大的热情对 NDN 进行研究,目的是满足当前用户对网络服务在内容传输和移动性方面的巨大需求和解决网络安全及性能方面的问题^[3]。目前 NDN 体系架构的研究重点主要分为命名数据结构、命名机制、安全问题、拥塞问题、转发策略和缓存策略。这些都是命名数据网络在设计之初重点考量的。在传统的 TCP/IP 体系架构中,在传输层面对拥塞问题作出反应,而路由转发层面则没有相关措施进行控制。NDN 网络与之不同,根据路由平面和转发平面的通告,智能转发,精准定位拥塞出现的位置,进而对拥塞做出响应。在 TCP/IP 体系架构中,采用的流量拥塞控制方案在 NDN 网络中并不适用,这是因为 NDN 网络的传输模式和 IP 网络不同,前者是用户端驱动的传输模式。数据是通过路由节点逐跳式传输,传输直接无连接,内容来源不固定。这些特性使得学者们要研究适合 NDN 的拥塞调节方案。本课题旨在分析传统拥塞调节算法存在的缺陷与不足,为网络拥塞控制性能提升制定相关的解决方案,缓解 NDN 网络中的流量拥塞问题,具有理论研究价值和实际应用指导意义。

1.2 国内外研究现状

20 世纪 90 年代末一些外国学者提出了以信息作为驱动进行数据传输的概念,并发表相关文章佐证理论,并且对内容分发和当前网络架构结合提出了见解,这是信息中心网络发展之初的研究工作^[4]。后来,对信息中心网络的研究步入了正轨,分别在网络架构和实际应用两方面展开。其中有关网络架构的研究包括数据命名、拥塞控制、缓存和转发方面的设计,实际应用方面则是根据特定的场景定制化设计,如物联网工程的通信、边缘网络接入等,这些都在一定程度上促进了信息中心网络架构的发展。

21 世纪初,NDN 获得美国国家自然科学基金的支持,由 UCLA 大学的张丽霞教授和 Van Jacobson 牵头,投入基金成立专项研究组,将 NDN 作为一个基础研究信息中心网络项目进行研究^[5]。在传统的网络体系架构中,用户向服务器请求内容,通过 IP 地址进行端到端的通信。NDN 与之不同,作为信息中心网络架构的代表,NDN 是以内容为核心的新型网络架构,内容取代了地址,成为网络中的首要实体。如何提升网络性能,缓解 TCP/IP 体系架构在地址有限性、扩展路由转发、拥塞控制等问题上的压力,成为 NDN 的研究重点。NDN 在很多方面对传统的 IP 网络进行了改进,例如命名机制,采用层次化结构,扁平化命名,对名字进行签名认证机制,保证内容的安全,避免出现内容污染;其次是通信模式,NDN 是接收端驱动的网络架构,网络中包含两种数据结构,分别为 Interest 数据

报文和 Data 数据报文, 用户端以封装内容名字的 Interest 数据报文请求内容, 内容源节点返回匹配 Interest 数据报文名字的 Data 报文响应; 在路由转发方面, NDN 采用面向内容分发, 对 IP 网络的地址解析和域内转发进行了改进, 并结合 NDN 路由节点可进行暂时缓存的特性, 缩短请求相同内容的往返时延, 提高传输效率。

从目前的情况看来, NDN 在新型网络架构体系中, 获得了巨大的成就和全新的突破。在目前的国际会议 SIGCOMM 和 INFOCOM 中^[6], 与 NDN 相关的研讨会已经举行了两次, 其中对 NDN 架构体系进行研究与实现的论文占比为 80% 以上。同时, 有硬件方面相关的研究, 第一台 NDN 路由器是由思科公司研制出来的, 并对其进行了公开演示, 获得一致好评。关于命名数据网络研究的官方组织也在不久后成立, 这一举措, 使得新型架构体系 NDN 从学术界走向了工业界。观望 NDN 网络的未来发展, 当命名数据网络在目前发展的基础上, 走出实验室, 在现实网络环境中应用时, 将会颠覆互联网体系架构以往的发展模式, 成为诸多技术中具有发展节点意义的一项成就, 而且, NDN 的投入生产将会缓解 IP 网络的压力, 消灭由 IP 网络带来的一系列痛点问题。

NDN 作为一种新型互联网体系架构, 研究学者在多个方面对其进行了深入研究, 比如命名机制、拥塞控制、安全机制、路由转发以及缓存策略等方面。相比于 IP 网络, 在拥塞控制中, 由于 NDN 的路由节点包含一段时间内的缓存, 支持多路径传播和多播传播, 增加了拥塞控制的复杂性。这点和 IP 网络不一样, 因为 IP 网络的端到端连接设计, IP 网络的拥塞控制是通过重复确认字符 (Acknowledgement, ACK) 和超时机制检测拥塞, 通过基于滑动窗口的和式增加, 积式减少 (Additive Increase Multiplicative Decrease, AIMD) 的方案减小客户端注入网络的流量^[7]。而由于 NDN 中数据包传播的多路径性, 每一个兴趣包请求的内容可能存在于多个不同的节点中, 这些节点在返回数据包时会产生不同的往返时延 (Round-Trip Time, RTT) 带来一定影响。尽管 NDN 和 TCP/IP 在传输模式上存在差别, 但是拥塞控制的本质, 在两种架构下是相同的, 都是在有限的网络资源下, 进行调度, 进而提高网络的传输性能, 总结起来就是预防拥塞的发生、感知并响应拥塞的发生。这两种网络架构都是根据反馈信息进行拥塞调节, 通过端节点和中间路由节点进行拥塞控制。这种拥塞控制的共性, 使得 TCP 的拥塞控制方法能够被引入到 NDN 中, 进而根据 NDN 网络的特点对算法进行调整。

由于学术界研究人员的不懈努力, 在命名数据网络的研究中取得了一些成果, 但由于 NDN 的还处于发展初期, 仍然缺乏相关理论和实验的完善, 在数据命名、

路由转发、数据缓存、安全隐私等方面，对 NDN 的研究尚待完善。由此可见，NDN 的研究前景广泛，需要学术界不同领域的研究者进一步完善和提升。

1.3 本文研究内容

虽然 NDN 网络以内容为核心，属于内容信息网络架构，但仍保留了 IP 网络的架构模型，表现为采用 IP 网络的沙漏模型，不同之处在于沙漏的“细腰”处，IP 网络的“细腰”处代表的是 IP 层，而 NDN 网络的“细腰”处则是内容部分。在内容块中，数据传输的核心是内容名字，内容名字是区别数据报文的标志，进而将 IP 网络的“从哪里”改为“是什么”^[8]。以此适应当前互联网中不断增加的内容量，解决 TCP/IP 架构中由于内容请求不断增加带来的痛点问题，对互联网的可持续发展产生了很大影响。

在任何网络架构中，都会出现网络拥塞的现象，网络拥塞本质是流量超过网络容量后发生的一种数据传输能力下降的现象。网络拥塞会对网络的正常传输造成很大影响，包括数据往返时延增加，内容请求得不到满足进而出现丢包现象，影响网络用户体验。虽然 NDN 网络能够解决 IP 网络中地址不足的问题，但对于日渐增长的用户数量和终端数量带来的网络需求的增大等问题，这些问题带来的传输压力使得研究学者们意识到在 NDN 中进行有效的拥塞控制至关重要。虽然目前对 NDN 的拥塞控制机制的研究已经小有成就，但是有些研究人员采用传统互联网架构中拥塞控制机制的方案来调节 NDN 网络的拥塞，并不完全适用于 NDN 网络，传统网络架构与 NDN 体系架构在结构上存在差异，传统的拥塞控制方案不仅无法发挥其优势，还会埋没 NDN 体系架构的优势，所以，开发适合 NDN 体系架构的拥塞控制机制尤为重要，这对 NDN 网络能否在未来网络体系中持久发展具有决定性意义。

本文的研究工作主要分为以下几个方面：

（1）网络拥塞检测机制的研究：

拥塞控制机制的首要工作是网络状态检测，目前 NDN 采用的拥塞检测方式总体分为三种，超时重传 RTO 机制、利用缓冲队列以及综合考虑数据往返时延与缓冲队列的方法。其中 ICP、CCTCP 等策略是超时重传 RTO 机制的代表，HoBHIS 等策略是后两者拥塞检测机制的代表。虽然目前的检测工作依赖这些策略，但是这些策略在网络状态还未到达拥塞的情况下，没有给出明确的网络状态结果。因此本文提出一种能检测网络空闲等还未达到拥塞状态情形的拥塞检测机制，实时检测网络状态，使得速率调节更加及时，网络带宽利用率更高。

(2) 基于用户端的拥塞控制方案的研究:

在 NDN 网络中,常采用 AIMD 算法进行速率调节,分为慢开始算法、拥塞避免算法、快速重传算法和快速恢复算法。还未发生拥塞时,发送窗口的增大规律表现为线性规律,也就是 AIMD 中的 AI (Additive Increase),当网络中发生拥塞情况时,慢开始阈值改变,变为发送窗口的二分之一,即乘性缩小 MD (Multiplicative Decrease)。由于实施本文提出的拥塞检测机制后,根据检测到的不同网络状态需要采取不同的速率调节方案,因此在用户端的拥塞控制方案改进了 AIMD 算法,根据网络状态针对性进行兴趣包速率调节。

(3) 基于路由接口排名的转发策略的研究:

由 NDN 网络的转发过程可知, FIB 结构中路由优先级字段对兴趣包选择转发接口起到一定的决策作用。将兴趣包尽可能从命中率高的接口转发出去,会提高路由节点的内容命中率,从而提高网络传输效率,从这个角度出发,本文设计的基于路由接口排名的转发策略提出了命中率优先级 (hit-rate preference) 的概念,并以此决定路由接口的排序。命中率优先级是路由接口转发的某一名字前缀的兴趣包在该接口发生命中的概率,命中率优先级与 NDN 中路由优先级字段的不同之处在于,命中率优先级概念依赖综合型拥塞控制算法的执行。整个过程为对网络进行拥塞检测后,获取到当前的网络状态,进而实时更新命中率优先级,选择命中率更高的路由接口进行转发。

1.4 本文结构

本文组织结构如下:

第一章,绪论。通过研究背景,简单介绍了当前互联网、NDN 网络的一些情况。结合国内外对于 NDN 的相关研究,对当前的研究现状进行了介绍与分析,在最后对本文的研究内容与组织架构进行了相应的介绍和总结。

第二章,NDN 基础架构及相关研究工作。本章分为三部分,第一部分围绕 NDN 网络整体架构设计展开,介绍了 NDN 沙漏结构、NDN 传输过程的数据包、数据包的命名、NDN 传输过程遵循的路由转发与缓存策略。后面两部分着重介绍了 NDN 拥塞控制策略和转发策略的相关研究工作,为后文中对 NDN 拥塞控制机制的优化提供了理论与实验基础。

第三章,综合型拥塞控制算法的拥塞检测设计。围绕网络拥塞控制机制,主要介绍了本文提出的综合型拥塞控制算法的设计,介绍了综合型拥塞控制算法中网络拥塞检测方案和具体的实施方法,包括实时监测网络状态,将网络状态划分

为空闲、繁忙、拥塞三种，并根据三种网络状态向链路中反馈不同的 NACK 消息，通告用户侧进行应对和调整。

第四章，综合型拥塞控制算法的速率调节设计。主要阐述综合型拥塞控制算法的速率调节策略设计，针对不同的网络状态，在用户端做出的兴趣包速率调整，不同于传统的 NDN 用户侧的速率调整算法，本文提出的基于接收端的速率控制更灵活，能更好的提高链路带宽利用率；除此之外，本章重点介绍了在中间节点的速率调节方案：基于路由接口排名的转发策略，本策略根据统计路由接口的兴趣包命中率，重新定义路由接口优先级，使得兴趣包尽可能的向命中率高的接口转发，进而缓解网络拥塞压力，提高网络传输效率。

第五章，算法的实验与分析。主要阐述本文所提出的综合型拥塞控制算法的实验实现。首先介绍实验平台，其次围绕实验平台 ndnSIM 的建立展开叙述实验的实现过程，最后在链路带宽利用率、路由节点内容命中率、用户总命中率三个评价指标方面对实验结果进行了对比分析。

最后对本文的研究内容进行了总结，阐述本文的研究成果，分析研究中存在的问题与不足，并提出了下一步可以研究的方向。

第2章 NDN 基础架构及相关研究工作

2.1 NDN 网络概述

2.1.1 NDN 沙漏架构

NDN 作为一种新型互联网架构，突破了当今互联网体系的局限性，并借鉴了目前互联网体系结构的成功之处。在 TCP/IP 中采用沙漏架构，沙漏架构通过上下层技术推动了网络的快速发展。传统互联网体系以实现全局互联所需的最小功能模块 IP 层为中心，如图 2-1 所示，将来自网络的数据存放在 IP 层，在 IP 层上层存放着应用产生的数据，在 IP 层的下层存放着结构互异的基础数据，这些数据的存放形式都是数据分组^[9]。随着互联网中用户数量的日益增长，网络中不断接入新的终端设备，位于中心位置“瘦腰”处的 IP 层，也面临由于 IP 地址与内容资源绑定带来的很多挑战。

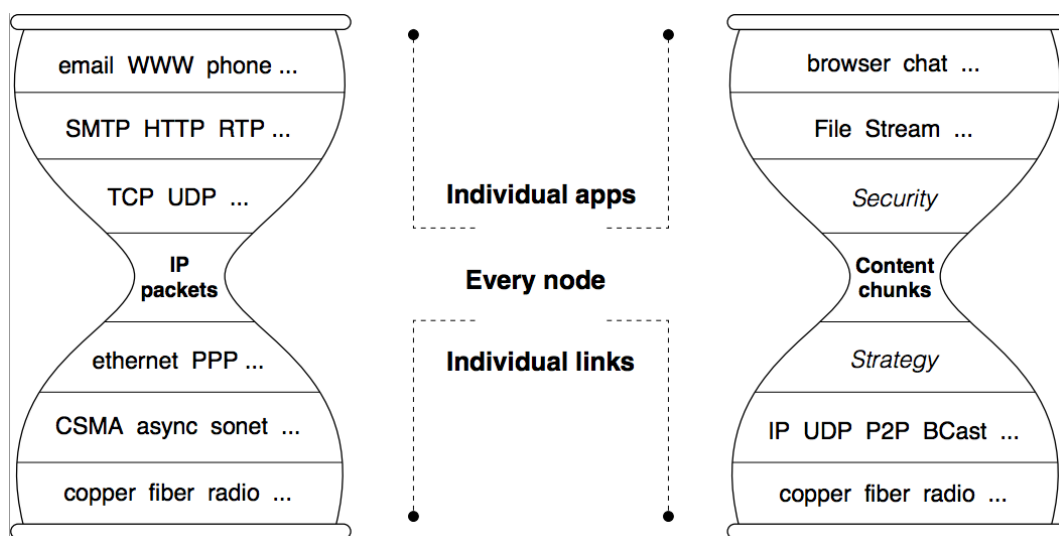


图 2-1 TCP/IP 协议栈与 NDN 协议栈示意图

Figure 2-1 TCP / IP protocol stack and NDN protocol stack

NDN 的设计并没有完全抛弃原始互联网体系中的沙漏架构，而是对原始沙漏架构做出了改动，如图 2-1 所示，“瘦腰”部位被内容块（Content Chunk）所填充，内容块取代了 IP 层，也就是在 NDN 网络中请求内容与 IP 地址之间不再有关联，取而代之的是名字前缀和请求内容的绑定，用户根据名字前缀寻求匹配内容。所以，在 NDN 的数据传输中，核心信息是请求数据的名称。由图 2-2 可以看到 NDN 网络通信与 IP 网络的不同。而且，NDN 的路由节点不仅具有计算

功能，还能进行存储，通过将网络资源缓存在路由节点上，节省了用户请求相同内容所需要的时间。因此，在网络内容资源和接入设备数量不断增加的背景下，相比于 TCP/IP 体系架构，NDN 采用内容块作为沙漏架构“瘦腰”的设计带来了诸多益处，在减少网内主要路径的传输压力，提高内容命中率，降低发生拥塞的概率等方面具有很大优势。

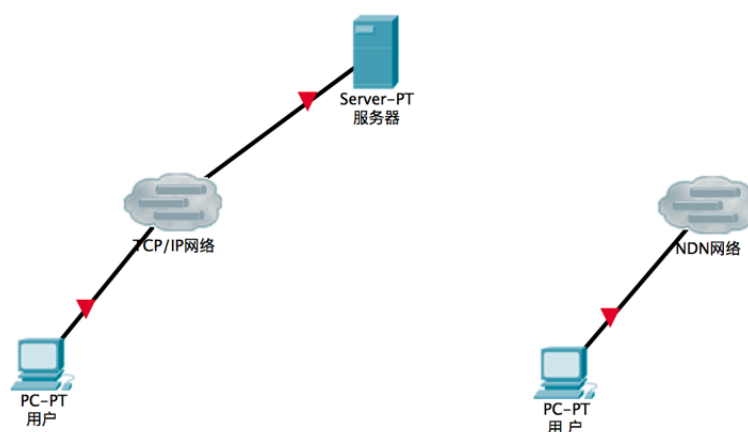


图 2-2 TCP/IP 与 NDN 的通信模式比较

Figure 2-2 Comparison of communication modes between TCP / IP and NDN

在 NDN 的网络架构中，安全层和策略层都作为独立的一层设置在网络模型中。安全层在网络通信过程中，提供了安全保障，减少了由于网络不稳定产生的意外攻击，进而降低了由于意外攻击造成的数据丢失等问题的影响。安全层中封装了不同的 Data 加密算法，在请求内容的数据分组中会附带内容提供者的签名信息，保证数据传输过程中内容的完整性，使数据内容在通信过程中保持可靠和安全。策略层指的是 NDN 网络的路由层面，即决定请求的传输方式。由于在网络传输过程中存在很多突发情况，例如用户请求量突然增大导致网络链路出现拥塞的现象，此时策略层发挥决策作用，指导请求的发送方式以及数据的返回方式，降低网络环境不稳定带来的意外影响。综上，安全层中设计的数据加密算法和内容验证方案让网络中的数据资源能够安全传输；策略层的转发策略以及对数据的管理操作为 NDN 的数据传输提供了功效方面的支撑。

2.1.2 数据包类型和数据结构

NDN 中的通信由接收端驱动。为了获取数据，消费者发出一个兴趣包^[10]（Interest Packet），其中标识了所需数据的名称，如图 2-3 所示，当兴趣包到达

包含请求数据的节点，该节点就会发送回一个数据包^[10]（Data Packet）。这个过程涉及到两种数据包类型，分别是兴趣包和数据包。这两种数据包都封装了表示请求内容的名字前缀，在网络的传输中通过名字前缀唯一识别^[11]。其中数据包中除了包含消费者所需的资源以外，还封装了用来验证内容安全的签名信息。

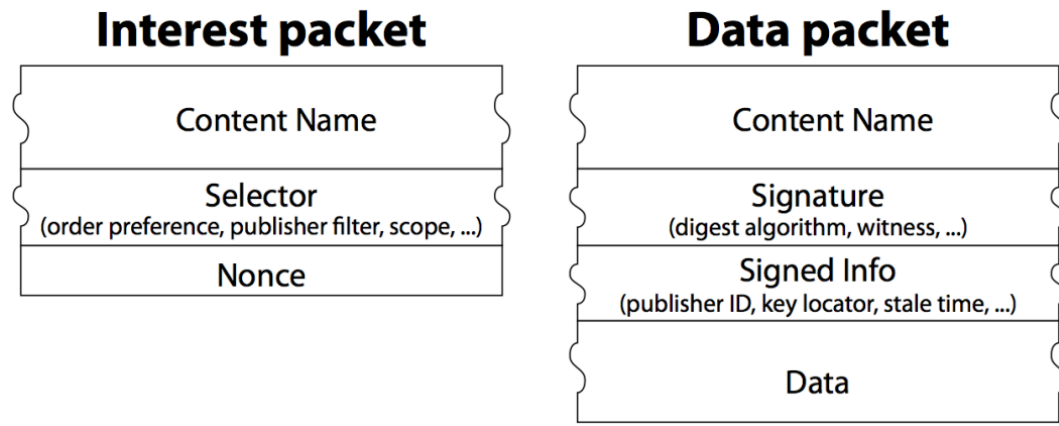


图 2-3 NDN 中 Interest Packet 与 Data Packet 的结构
Figure 2-3 Structure of Interest Packet and Data Packet in NDN

NDN 的兴趣包包含内容名字、选择器和随机字段 Nonce。兴趣包的传输过程是单向的，因此避免了环路传输的可能，这是 NDN 多路径传输的基础^[12]。Nonce 字段的设计就是用于检测重复的兴趣包，当出现相同兴趣包时，根据随机数 Nonce 将重复的兴趣包删除。当兴趣包超过生存周期时，也会被丢弃或重传，这个生命周期默认为 4s^[13]。与兴趣包不同的是，数据包的传输是按照兴趣包转发的路径原路返回，不会出现环路，所以没有 Nonce 字段进行重复检测。

NDN 的内容数据包是响应兴趣包请求的，数据包中包含内容名字、签名字段和数据内容。签名字段用于保证数据传输过程中的安全性，这是沙漏模型中安全层的宗旨^[14]。用户通过验证数据包中的 Signature 字段，就可以识别出返回的数据是否安全，确保内容没有受到污染。而用户请求内容的正确性，则是通过验证 Signature Info 部分来确认的^[15]。Signature Info 这部分信息包含了一系列加密算法，通过这种机制，确保 NDN 在传输过程中的安全。

用户想要获取资源时，会通过发送兴趣包的方式获取，在兴趣包中封装了请求内容的名字。然后通过网络的路由节点转发，当转发至某一节点，若该节点包含用户所需内容，则由该节点返回相应的内容数据包，传输到用户端。至此，用户的请求得到满足^[16]。在 NDN 中，内容数据包的返回不需要转发策略的指导，它的返回路径是兴趣包转发路径的相反路径。这样的转发过程与传统的互联网传输方式是不同的。

介绍完 NDN 的两种数据包类型，下面介绍 NDN 中在通信过程中起到至关重要的作用的数据结构：转发信息表（FIB, Forwarding Information Base）、待定兴趣表（PIT, Pending Interest Table）和内容缓存表（CS, Content Store）^[17]。

FIB 与 NDN 中兴趣包的转发策略有关^[18]。和 IP 网络中的转发信息表相同的是 NDN 中的转发信息表同样是指导网络转发的依据，不同的是 NDN 的 FIB 中保存的是内容名字前缀，并且为这些名字前缀匹配了相应的转发端口，它们是根据路由协议生成的转发兴趣包去内容源方向的接口，并且一个名字前缀可以对应多个可以转发的端口。

PIT 中记录了兴趣包的来源与去向，方便返回的数据包按兴趣包转发来的路径原路返回^[19]。PIT 中的每一条记录都置有与内容相关的信息、兴趣包的进出接口列表、所有接收到的相同名字兴趣包的最大生存时间以及以转发策略标签形式存在的转发信息。PIT 使用唯一的内容前缀记录兴趣包的信息。当遇到相同的兴趣包时，在 PIT 中记录该兴趣包的进接口，使得相同的兴趣包得到聚类。当数据包返回时，按照 PIT 表中记录的映射关系，沿着兴趣包转发过的路径原路返回，最终到达用户端。

在 CS 数据结构中，存储着网络中的内容资源^[20]。由于缓存替换策略的实施，这些数据能够在 CS 中存放一段时间，在这段时间内用户发送的请求可以通过包含内容的 CS 所在的路由节点满足，用户不必再去内容源获取数据。通过在路由节点采用不同的缓存策略，动态调整内容的缓存时长，即热门内容增加缓存时间，冷门内容减少缓存时间。考虑到资源开销问题，访问次数较少的冷门内容会被路由节点从缓存中删除。

2.1.3 命名规则

在 NDN 的数据传输过程中，网络架构模型的每一层都能够获知数据包中封装的名字前缀，名字前缀根据请求内容的特征制定，同时也会采纳请求方的自身需求^[21]。在 NDN 网络中，携带名字前缀的数据结构为兴趣包、数据包以及与兴趣包同名的 NACK 消息包，名字前缀在这三种数据结构中都是以二进制形式封装的^[22]。NDN 的数据包内容名字与内容所在的地址无关，这一点与 IP 网络是不同的。此特性也契合 NDN 网络多源多路径的特点。NDN 通信时，包含请求内容的数据包每到一个路由节点，该路由节点会将此数据包的内容暂时缓存，并在遇到相同内容请求时，返回其缓存的数据^[23]。NDN 的命名采用层次结构，将内容名字分成不同的层次，不同层表示不同含义，例如

“/bjut.edu.cn/vedio/vedio.mpeg/s1”，其中将请求内容的名字划分为四个层次，如图 2-4 所示，“bjut.edu.cn”为可路由前缀，“vedio”代表内容的类型为视频，

“vedio.mpeg”表示视频的名字，“s1”是视频分段传输名字。设计这样的分层式命名结构能够清晰的显示所要请求的内容类型和大小，提高了数据包的传输效率，有利于提升网络的整体性能^[24]。

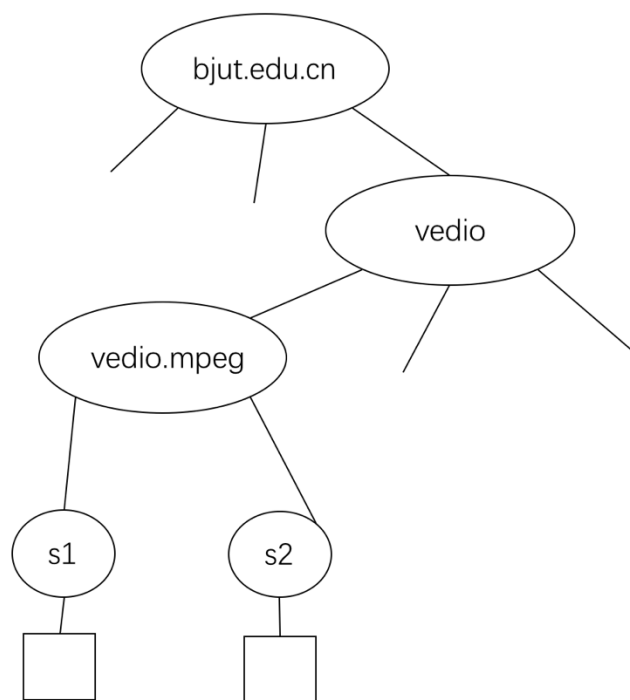


图 2-4 NDN 层次命名结构

Figure 2-4 NDN hierarchical naming structure

随着科技发展的日新月异，互联网中用户规模不断扩大，随之而来的是用户产生的海量数据，进而使网络中数据更新迭代加快，如何高效处理越来越多的数据，在数据的命名方面面临很大挑战^[25]。因此研究学者制定命名规则时需要考虑很多问题，首先为了提高内容传输效率，NDN 网络会采取分组请求的方式，将完整数据分成几个部分分开请求，进而使数据分组返回；其次由于缓存替换策略的实施，路由节点会删除网络中访问次数较少的数据，这部分数据可能是分组请求内容中的一部分。所以为了维持 NDN 网络的命名秩序，需要制定一些命名原则^[26]，首先要遵循命名是为了更好的进行数据传输，保持网络正常通信的状态，避免由于网络不稳定出现意外导致传输中断，其次为数据命名时，应考虑到 NDN 路由节点能够缓存的特性，实现在内容命名层面提高数据响应时间，进而提高传输效率的效果。最后通过设计合理的命名机制，能够更好的实现 NDN 的多路径传输。

2.1.4 路由与转发

通过不断的将内容名字和网络资源进行匹配就实现了 NDN 网络的路由转发, 这是一个在 FIB 中建立映射的过程。采取最长前缀匹配的方式将兴趣包的名字前缀与 FIB 表项进行匹配, 确定兴趣包的转发方式。NDN 中路由转发过程如图 2-5 所示,

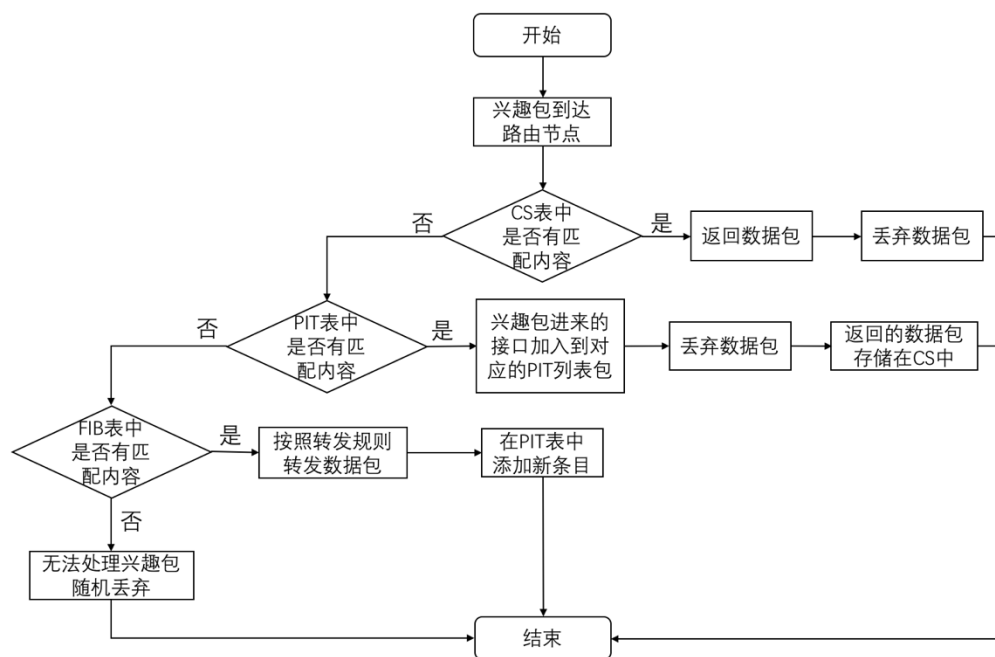


图 2-5 NDN 路由转发过程

Figure 2-5 NDN route forwarding process

路由算法决定兴趣包的转发路径, 在 NDN 中, 借鉴 IP 网络中的链路状态路由算法和距离矢量路由算法, 完成对转发路径的决策。下面介绍目前学者们对路由算法的相关研究, 在网络多路径转发中, 事先获取网络中各路径的链路状态, 就能够为转发路径的选择提供很好的决策支持。而开放式最短路径优先协议 (OSPF)^[27]就是如此, 通过将链路状态报告 LSA 发送给一定区域内的所有路由节点, 告知各路由节点目前的网络链路状态。基于此研究人员在 NDN 网络上实现了 OSPFN 协议^[28], 并验证了其有效性。由于 NDN 网络通信需要将请求发送到多个路径中, 以便提高内容的命中率, 而 OSPFN 并不能动态的选择多个路径进行转发, 所以命名数据链路状态路由协议 (Name-data Link State Routing protocol, NLSR)^[29]应运而生。该协议区分链路和路由的方法是通过使用名字前缀在网络中传输, 而各路由节点构建符合传输需要的拓扑结构, 并计算出最短路径, 然后以此对可用接口排名, 并将接口排名封装到 FIB 表中, 生成多路径路由表, 以此方法实现多路径的动态转发。

研究人员通过对 NLSR 的模拟测试,发现 NLSR 在实际应用中仍存在一些隐患,例如在网内内容变化次数增加时,NLSR 没有及时更新转发路径的变化,并且在运行过程中内存开销大。在图 2-6 所示的网络中,路由节点 A 和 B 距离两个内容源较近,分别收到来自两个内容源的消息,消息通告内容源所包含的数据。节点 A 和 B 收到之后,更新转发信息表,并向网络中通告内容的名字前缀,直到网络中所有节点都接收到内容的状态信息。通过对这个例子的分析可知,同步机制对 NLSR 的使用效率产生直接影响,因此如何更好的优化 NLSR 的同步机制是今后的重点研究方向。

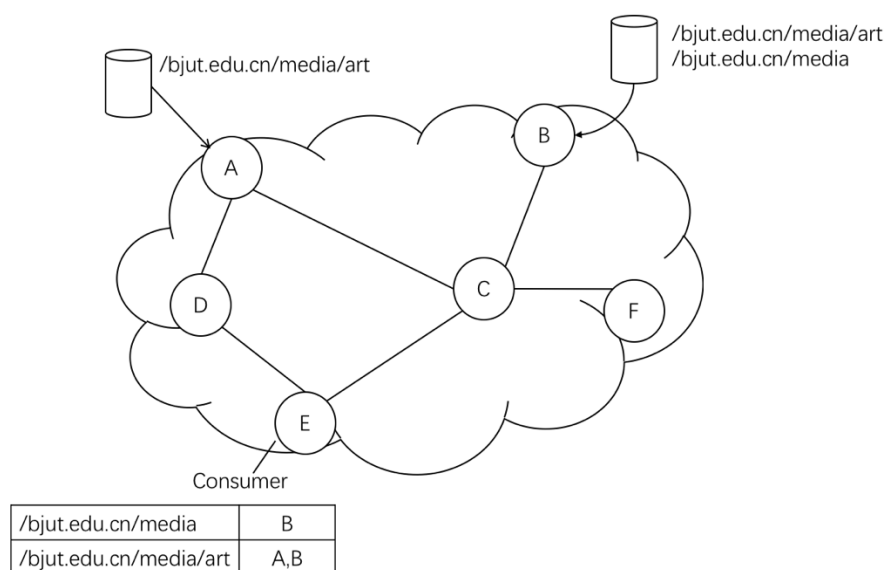


图 2-6 NDN 路由机制

Figure 2-6 NDN routing mechanism

NDN 中的路由节点和其内部结构在路由转发机制中起到了关键作用,图 2-7 描述了 NDN 路由节点的内部结构。汇总 NDN 的路由协议、转发信息表、内容存储表和待定兴趣表的数据信息,NDN 体系架构构建了路由节点的转发模型,这种转发模型综合链路状态信息,调整为适合每个路由节点的模型,通过这个转发模型决定内容请求的转发策略,例如路由器转发兴趣包的进接口和出接口,待定兴趣表中记录的兴趣包前缀数量,兴趣包的多路径转发产生的网络流量控制,网络拥塞发生后兴趣包该往哪些接口转发等问题。同时,NDN 转发策略关注待定兴趣表中记录的兴趣包与返回状态信息,这是 NDN 在转发中保持正常通信的基础。

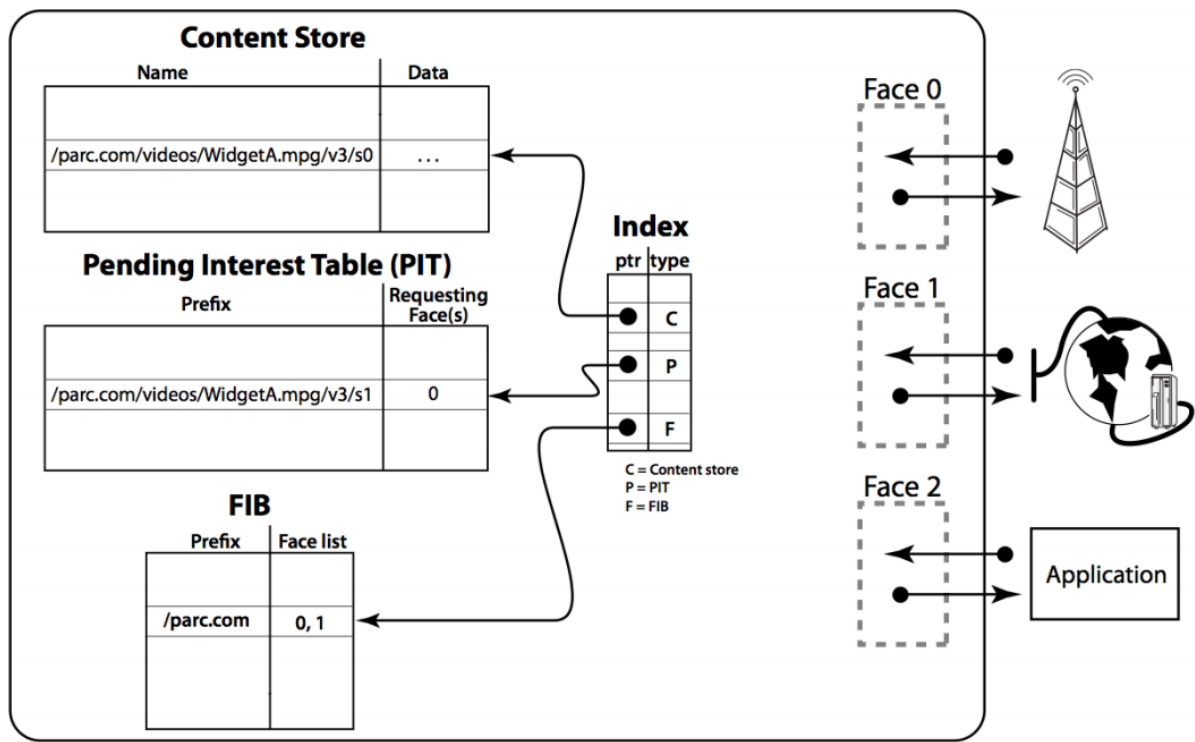


图 2-7 NDN 转发节点的内部结构

Figure 2-7 Internal structure of NDN forwarding node

综上，在 NDN 的传输过程中，通过获取基础数据结构转发信息表和待定兴趣表中记录的状态信息，使得 NDN 在转发过程中更加灵活和智能，能够更好的适应多路径传输，更高效的控制网络流量。

2.1.5 缓存机制

NDN 中路由节点具有存储功能，为网络中错误数据的修正和通过不同节点返回数据提供了功能保证。此外在 NDN 体系架构的沙漏模型设计中，“瘦腰”部分为内容块，同时分层式的命名结构，使数据内容与地址没有关联，也就使得在兴趣包的转发过程中内容名称的权重高于内容地址的权重，因此，请求相同内容的数据可以在网内中间节点的缓存中得到，从而缩短数据往返时延，减少网络中过多的传输流量，从而提高了网络传输效率。

在 NDN 网络中设置了路由节点的缓存机制，缓存机制可以降低网络流量、提高用户端的响应时间，所以缓存机制的重要性不言而喻。NDN 网络的缓存机制主要有两种：分别是缓存替换机制和缓存决定机制。这两者的区别在于，缓存决定机制的本质是保留信息，而缓存替换策略则是删除信息。下面对两个缓存策略进行详细的介绍，当用户端向服务端发起请求后，服务端会将响应请求，并将请求内容沿兴趣包传输路径逆向返回给用户端。为了防止服务器多次响应相同的

请求,缓存决定策略将在响应内容返回的路径上,选择合适的路由节点作为响应数据的缓存节点,从而缩减了用户端请求的响应时间以及降低网络中的通信开销。而随着路由节点中缓存的信息越来越多,不得不扔掉一些信息的时候,缓存替换策略就起到了重要的作用。缓存替换策略会通过判断将一部分缓存价值低的信息删除,同时更新一部分有缓存价值的信息。虽然缓存决定策略与缓存替换策略相比,缓存替换策略的设计对 NDN 数据传输的效率影响更为直接,但是只有两者共同协作,才能更好地为 NDN 网络服务。

2.2 NDN 拥塞控制策略的相关研究

网络拥塞本质上是网络资源供需关系的协调问题,具体来说,就是网络无法处理超过资源上限的内容时发生的情况,被称为网络拥塞。如何有效解决网络拥塞,使数据正常传输是目前互联网发展中亟待解决的重点问题。目前采用较多的拥塞控制方案,有以下三种:在用户端的拥塞控制策略,逐跳的拥塞控制策略和混合控制策略^[30]。

在用户端实施的拥塞控制策略采用计数器超时策略判断网络是否发生拥塞,具体的拥塞控制措施是通过减少用户端发送兴趣包的数量,从而减少返回的数据包的数量,以此来解决网络拥塞。图 2-8 描述了用户端进行拥塞控制的原理。图 2-8 的左边表示拥塞窗口增大的情况,右边表示拥塞窗口减小的情况。

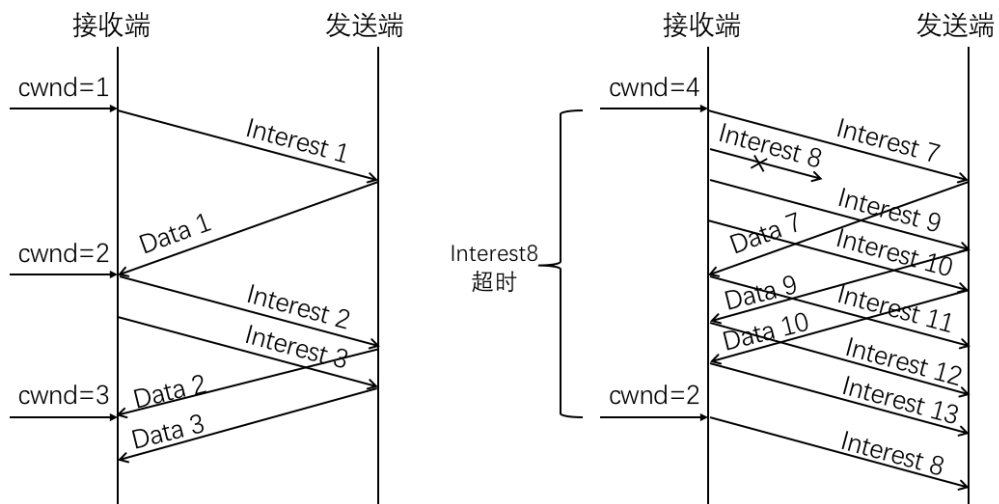


图 2-8 基于用户端的拥塞控制
Figure 2-8 User-based congestion control

如图 2-8 所示, 左边拥塞控制窗口 (cwnd) 会随着兴趣包在网络中的转发发生变化, 当兴趣包对应的数据包返回后, 拥塞窗口会增大。当图 2-8 右边触发 RTO 计数器超时后, 拥塞窗口减小, 在用户端, 会重新发送未被满足的兴趣包。以上这个过程和传统的 TCP/IP 体系架构存在差异: 首先, 在 NDN 中, 数据包的传输非有序, 这是因为 NDN 采用前缀匹配的方式将兴趣包和数据包对应起来; 其次, 由图中可以看出, 每个兴趣包转发到一个节点时, NDN 会对其确认一次, 这与 IP 网络中的不间断累积确认是不同的; 最后, 出现丢包现象时, NDN 会发送下一个兴趣包匹配的数据包, 不会出现 IP 网络中收到相同内容的情况。

由于 NDN 的通信过程不同于 IP 网络, 因此, 将 IP 网络中的拥塞控制方案直接搬到 NDN 网络中使用存在不兼容的现象。具体来讲, NDN 中路由节点可以暂时性的缓存内容资源, 这使得我们在计算数据往返时延时, 得到的结果并不准确; 其次, 在用户端通过调整兴趣包的发送速率来进行拥塞控制的措施是一种被动的调节拥塞的机制, 这样的方式虽然会减少发送兴趣包的数量, 但用户端仍然在发兴趣包, 这种行为很可能会增加链路的负载, 致使拥塞情况变得更加严重。

由于命名数据网络中传输特性具有多源性和多径性的特点, 因此拥塞控制研究在考虑到这个特点后, 提出了单径拥塞控制和多径拥塞控制的方案。

首先介绍单径拥塞控制的研究工作。基于用户端的单径拥塞控制策略的典型代表是 ICP (Interest Control Protocol) [31], ICP 是最早应用到命名数据网络中的拥塞控制机制之一, ICP 沿用了 AIMD 算法, AIMD 算法广泛应用于传统互联网架构的拥塞控制中。具体来讲, 在 ICP 中, 发送出去的兴趣包会携带一个超时重传时间, 而实际网络中数据传输的往返时间也会被记录, 然后将这两个超时重传时间进行比较, 通过这样的方式进行网络检测。当请求的资源没有在超时重传时间内返回到用户端, 就认定发生了网络拥塞。进而采取拥塞控制方案, 即降低兴趣包的发送速率, 从而降低数据包的传输速率, 达到调节网络拥塞的目的。由以上过程可知, 当网络中没有传输较多数据时, 在用户端的单径拥塞控制机制采取 AIMD 算法进行拥塞控制, 可以实现及时检测和缓解局部偶发性拥塞。

然后介绍多径拥塞控制的研究工作。在命名数据网络中, 一个内容请求处发送的兴趣包请求分组可以通过多条路径转发, 并且可以通过多个内容源响应数据。不同的传输路径带宽不同, 当在用户端进行网络状态检测时, 若检测到发生拥塞, 很大可能只是某一条链路上发生了拥塞, 此时其他路径上未必产生拥塞, 此时, 单径拥塞控制策略统一降低网内路径的兴趣包发送速率, 从而造成某些处于正常传输状态的路径带宽利用率降低, 影响整体网络传输效率。CCTCP (Content Centric TCP) [32] 为每一条兴趣包请求的数据流和内容返回的数据流都设置了发送速率和重传时间, 这一点与 ICP 是不同的, 这样做的目的是解决上述单径拥塞

控制策略出现的问题，在 CCTCP 中，不同的数据流根据响应请求的路由节点区分，由于在命名数据网络中数据传输时是以内容名字为核心，对路由节点的位置并不关心，所以满足请求的路由节点是通过比较传输内容的往返时间区分的。CCTCP 策略在 NDN 的多径性方面优于 ICP，但是由于 CCTCP 需要维持多个路径的超时重传及时器，这使得网络中出现了不必要的工作量，而且异地缓存数据功能的存在使得计算数据往返时延较为困难，CCTCP 在多径性的数据传输网络中也存在明显缺点。

除此之外，还有一些学者在基于用户端的拥塞控制方面做出了自己的研究。Arianfar S 等人^[33]从 NDN 的传输协议方面考虑，借鉴传统的 IP 网络中拥塞窗口的概念，提出了一种新的传输协议。在该协议中，通过检测内容数据包的数量来决定滑动窗口的变化，设计新的转发机制进行数据包的分发与响应。这种新的转发机制要求用户端和内容源存储了对方的信息。这种传输协议充分考虑了 NDN 体系架构的特性：基于接收端驱动的数据获取方式。即在接收端发送内容请求到网络中，从而获取想要资源的方式。通过 NDN 这种特性，作者提出的传输协议就可以实现通过控制兴趣包的发送速率，进而控制数据包的返回数量。所以利用基于 TCP 窗口的机制可以很好的控制网络中内容数据包的流量。在此基础上，Salsano S 等人^[34]进一步研究，通过控制内容数据的大小，来解决安全机制和数据流控之间的协同问题。在 NDN 体系架构中，网络的带宽大部分被数据包所占据，因此可以说数据包的数量决定了网络是否会发生拥塞，基于此，Amadeo M 等人^[35]提出的拥塞控制算法通过统计各个数据包返回的时间间隔，控制兴趣包的发送的时间间隔，和 IP 网络的拥塞控制算法相比，作者提出的算法在拥塞控制上效果显著。文献[36]中仍沿用 TCP/IP 架构中的 RTO 超时机制，与 IP 网络不同的是，作者为每一个内容提供节点设置一个 RTO 计数器，这样做的前提是内容提供节点的位置和网络传输方式保持不变。这种方式在 NDN 体系架构中实现起来非常困难，且有一定的局限性。在这种思路的基础上，很多研究者将计时器设置在 NDN 其他的数据结构上，文献[37]的作者为 NDN 的内容缓存表 CS 设置计时器，并在中间路由节点中封装唯一内容缓存表标识，这些标识用于标记所有经过该路由节点的内容数据包，这样 CS 缓存过哪些数据包就非常清晰。在用户端以 CS 为单位，配置 RTO 计数器，从而建立内容数据包与 CS 之间的对应关系，这样就能得到请求内容的数据往返时间。虽然该方案在拥塞控制方面取得了一定效果，但是由于缓存替换策略的存在，使得数据往返时延发生变化，该方案并没有考虑这一点带来的数据更新问题。

逐跳的拥塞控制机制是一种主动避免拥塞的机制，在命名数据网络中，每一个路由节点都维护了一个待定兴趣表 PIT，该表记录着经过此路由节点的 Interest

数据包和 Data 数据包的信息，同时 Interest 数据包的转发决策基于该表，因此在命名数据网络的传输中，可以在路由节点进行网络状态检测和拥塞控制。

图 2-9 显示的是基于路由节点的拥塞控制过程，在中间节点 R_i 的传输过程中，路由器会根据经过该节点的数据包数量、数据流的时间间隔、CS 中内容的存量变化、网络的实时带宽等信息，判断接口的状态。通过接口的状态判断网络状态，然后路由节点对兴趣包转发接口进行排序，并作出相应的速率调节措施。

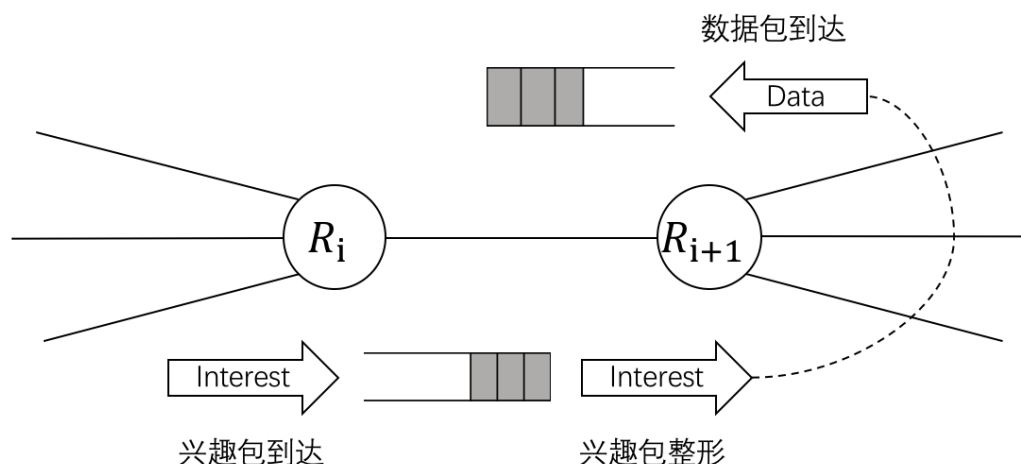


图 2-9 基于中间节点的拥塞控制

Figure 2-9 Congestion control based on intermediate nodes

基于此，在 NDN 的研究中，掀起一股对在中间节点进行拥塞控制的研究热潮。Carofiglio 等人^[38]将路由节点的传输队列长度，链路带宽和响应时延这些数据统计起来，计算每一跳的 Interest 数据包的发送速率，当网络中出现过剩的兴趣包时，主动丢弃，这样的操作减少了网络中一部分流量，从而减轻了网络拥塞的压力。但是在该策略的实施过程中，每个数据流都会得到相同的队列缓存空间，这样虽然保证了数据流之间的公平性，但是当数据流不需要这么多缓存空间时，会造成不必要的网络资源开销。文献[39]是基于命名数据网络的逐跳流平衡特性，考虑 Interest 数据包对网络状态的影响，根据分析出的场景：兴趣包的数据流和数据包的数据流是对应关系。问题是优化这种双向数据流的传输效率。基于这样的场景和问题，作者设计了一种逐跳的兴趣包（Hop-by-hop Interest Shaping, HIS）整形方案。HIS 方案根据设定的链路带宽，通过路由节点收到的数据包数量决定发送兴趣包的速率，避免由于传输路径的差异和兴趣包与数据包之间大小的差异对链路传输效率产生的影响。然而这种策略也存在弊端，当用户请求的内容在网络中广泛存在时，返回的数据包数量会多于其他内容请求返回的数据包，这样就造成了数据流之间的不公平传输。文献[40]从内容访问次数角度考虑，通过统计网络中内容的访问次数，预测未来哪些内容会变成被访问较高的内容，进而在路

由节点调整兴趣包的发送速率，从而提高缓存命中率，减少冗余输出。该方案降低网络开销方面取得了一定效果，但由于内容访问次数的不断变化，如何对其进行维护有待更深层次的探索。文献[41]引入深度学习概念，通过深度学习的神经网络结构进行预测，将预测结果作为指导来实施相应的转发策略，从而避免发生网络拥塞。然而设计合理的神经网络模型和训练复杂度低的模型成为当前亟待解决的问题。

最后是将基于用户端的拥塞控制机制和逐跳的拥塞控制机制结合，图 2-10 说明了这种混合拥塞控制策略的原理。在实施混合策略的网络中，所有路由节点都使用的一套拥塞检测机制和拥塞控制机制。通过用户端的被动控制和网络中路由节点的主动控制综合，实现更高效的网络拥塞控制和资源分配。

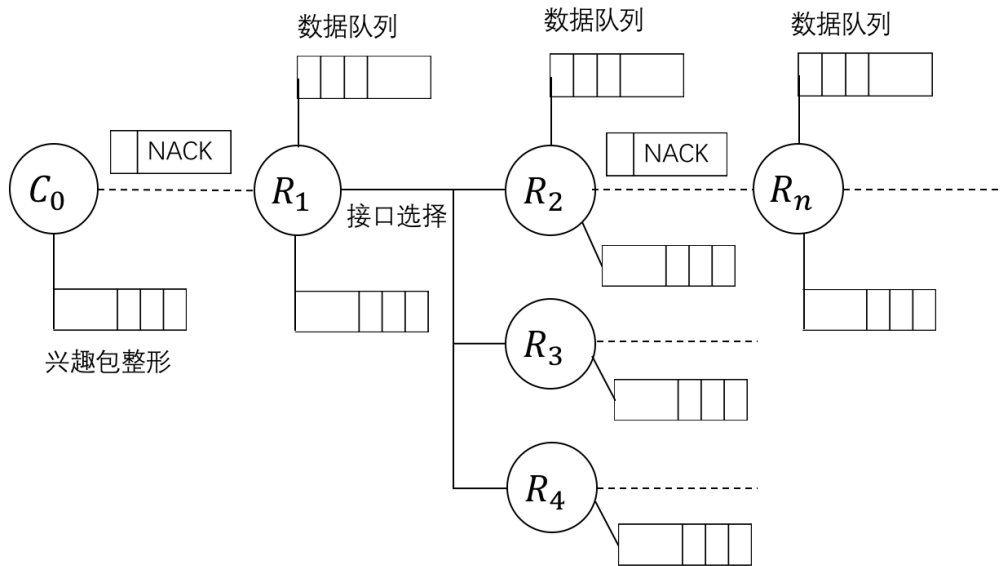


图 2-10 混合拥塞控制模式

Figure 2-10 Hybrid congestion control mode

文献[42]通过考虑用户侧和网络侧两方面，设计了一种混合拥塞控制机制，有效的实现了快速拥塞响应和资源高效分配。在NDN网络中兴趣包的转发会因为多源多路径而产生拥塞，为了更好地对拥塞进行控制，Chen, Qingxia等人^[43]在将用户端和路由节点的拥塞控制方法结合之后，还加入了随机指标标记算法，用以更直观的展示网络拥塞的状态。

文献[44]设计出一种实用拥塞控制策略，针对网络中实际可用的链路带宽和数据包的大小等一些问題，采用 CoDel^[45]方式主动对数据包进行标记，以此作为检测网络状态的指标。进而来调节各个路由端口的兴趣包转发概率，并在用户端调节兴趣包的发送速率。此外，该策略中条件限制因素减少，因此能够应用到多种场景中。文献[46]设计了一种在用户侧使用 ICP 进行拥塞控制，在中间节点维护虚拟队列的拥塞控制方案。该方案提出在中间路由节点的转发接口上，为每个

数据流设置一个虚拟队列，除此之外，还对虚拟队列进行信用评估，通过统计数据流传输的字节数，将引起拥塞的数据流定义为瓶颈流，与这样的数据流对应的兴趣包信用计数会降低，直到信用降为 0 时，就判定转发该兴趣包会产生拥塞，进而将其加入需要进行速率调整的兴趣包队列。

2.3 NDN 转发策略的相关研究

在 NDN 网络设计中，能够实现网络内缓存、多路径传输等功能，为了使缓存资源得到有效利用，很多学术界研究人士对命名数据网络的转发机制进行了深入研究，并取得了很大的突破。

文献[47]提出最佳路由转发策略（Best-Route）策略将兴趣包转发到网络开销最小的下一跳路由。具体步骤是通过 Dijkstra 最短路径算法计算出相应条目的最短路径，并在 FIB 表上安装相应接口。虽然该策略将最小开销考虑在内，但是由于在单一路径下，兴趣包无法实现自适应转发，随着网络中流量升高，兴趣包的发送速率增大，使得网络链路会出现带宽不足的情况，进而出现转发性能降低的情形。

Afanasyev 等人^[48]提出了一种基于颜色的转发策略，用三种颜色表示转发接口不同的状态，分别是“绿”代表接口正常，“黄”代表接口就绪，“红”代表接口故障，如表所示，

表 2-1 转发接口状态与对应颜色表
Table 2-1 Forwarding interface status and corresponding color table

接口状态	颜色
正常	绿
就绪	黄
故障	红

该策略整合处于同一状态的接口，并对这些接口进行排序，排序的依据是接口的响应时间。在三种颜色对应的接口中，兴趣包会被优先转发到绿色和黄色的接口处，即绿色和黄色对应的接口优先级高于红色接口。Carofiglio G 等人随后提出三种转发策略，分别是洪泛策略、最优路径策略和最小时延策略，洪泛策略将兴趣包转发到所有可用接口，最优路径策略将兴趣包转发到路由机制给出的最短路径接口，最小时延策略顾名思义将兴趣包转发到最小响应延迟接口。这三种转发策略的实施都依据路由信息，而在实际网络中，保持实时更新路由信息会花费很大的网络开销，并且付出巨大的时间代价。

Yeh 等人^[49]提出了虚拟兴趣包的概念，以在虚拟网络平台上通过背压机制获取的兴趣包转发与缓存策略为基础，为现实中的兴趣包的转发与缓存提供了参考，但该策略的缺点是在实际应用中的复杂性高，同时增加了路由器之间信息传递的消耗。

A. Udugama 等人^[50]提出了一种按需多路径转发策略。该策略提出内容请求方可以根据自己的需求，在用户端通过多个端口发送兴趣包，然后根据请求内容回到转发端口的时延，来调节该端口发送兴趣包的速率。除此之外，这种策略还设计了一种避免网络中出现环路的方案，即每个端口只能发送请求相同内容的兴趣包。虽然这种方案在一定程度上避免了网络出现环路的现象，但是每个端口只能发送请求相同内容的兴趣包，使得网络中的带宽资源无法充分利用。

Andrea Passarella 等人^[51]提出 NCC 策略，该策略将兴趣包转发到数据包延迟最低的接口。具体步骤是先设置数据结构表用来收集 PIT 的条目信息，然后挑选一个接口作为最佳接口，最后调整预测延迟使得选出的接口能提供最低延迟，并且不断更新最佳接口。虽然 NCC 将延迟最小考虑在内，并且在发生拥塞时改变转发接口。但是 NDN 的网络架构形式为以信息为中心，主要功能为网内缓存，该策略中由于端口的转换可能导致兴趣包无法匹配到包含内容的节点。

Yeh 等人^[52]在 NDN 的基础上，提出了一种集成缓存和动态转发的网络架构。这种网络架构中通过两种机制来指导数据的传输，分别是虚拟的转发指导策略和实际的转发指导策略。虚拟的转发指导策略的设计的关键是满足用户的所有需求，即转发用户提出的所有兴趣包，当然这些兴趣包是虚拟的。而实际的转发指导策略直接转发网络中的兴趣包和数据包。虚拟的转发指导策略的作用是指导实际的路由转发过程，通过与分布式算法结合，由此产生的流率和队列长度是指导实际转发的关键。如图 2-11 为该网络的架构。虽然该转发策略在网络成功传送数据方面取得了一定的效果，但是由于没有考虑转发端口的状态，因此实验结果可能出现偏差。

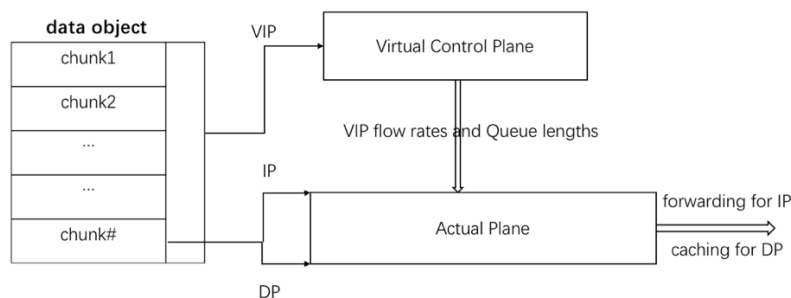


图 2-11 VIP 架构

Figure 2-11 VIP architecture

文献[53]在 A. Udugama 等人的研究基础上, 将这种按需分配的转发策略分成两部分, 首先将网络环境设为包含多条传输路径的结构, 在这些路径中, 网络资源分别缓存在不同的路由节点中; 其次提取每条路径的属性特征, 依据这些特征指导兴趣包的转发。图 2-12 说明了实施该策略时名字前缀为 /unibremen/coments/cn1/lect.avi 的请求的转发过程。

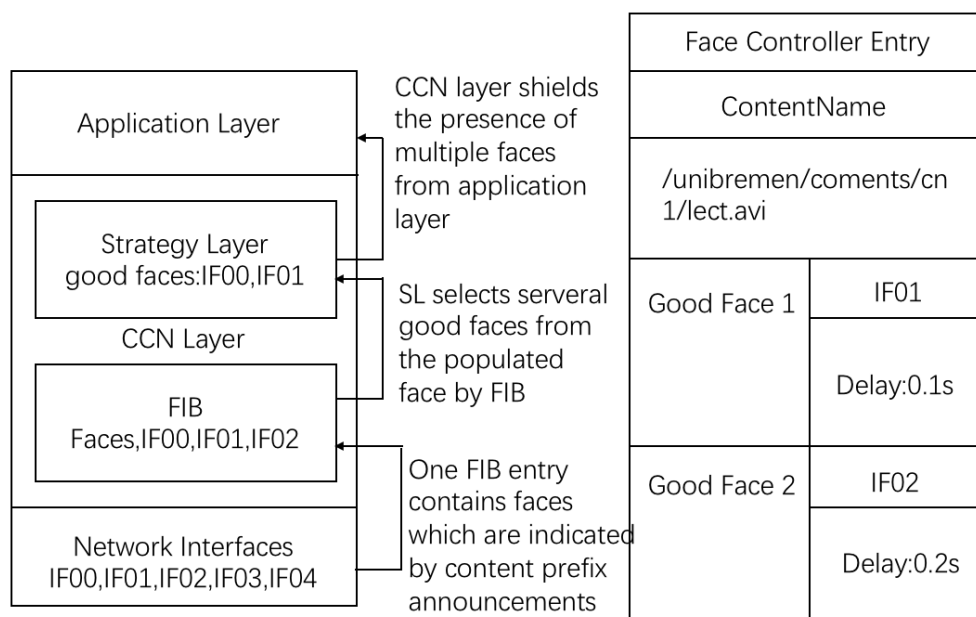


图 2-12 基于路径特性的转发流程

Figure 2-12 Forwarding process based on path characteristics

路径的属性特征是实施该策略的关键指标, 而评估路径的属性特征的是路由转发接口的时延。接口时延决定该接口转发兴趣包的概率, 并且以线性规律实时更新。虽然这种策略理论上能够通过评估转发接口时延获知网络中内容资源的位置, 但在实际实施的过程中并不一定能达到预计的效果。这主要是由于 NDN 中路由节点会执行缓存替换策略, 这一策略的执行会替换掉节点中缓存的某些内容, 这样以来, 接口时延就不再准确。而且, 策略以线性规律更新接口时延, 并不能实时反映网络的状态, 导致在路由节点调控兴趣包发送速率的措施也会延迟。

2.4 本章小结

本章首先对 NDN 网络架构进行概述, 包括 NDN 协议栈、基础数据结构、NDN 的命名策略、路由与转发策略以及缓存策略。然后对现有转发策略和拥塞控制策略的研究工作展开了描述。为之后对 NDN 网络的优化进行了铺垫和准备。

第3章 综合型拥塞控制算法的拥塞检测设计

通过上一章中对 NDN 拥塞控制相关研究的总结得出,基于用户端的拥塞控制和逐跳的拥塞控制机制成为目前主流的拥塞控制算法。这两种拥塞调节算法,考虑了 NDN 体系架构的特性,实现了能够发挥 NDN 数据传输特性优势的拥塞检测与控制算法。虽然在拥塞控制上取得了一定的效果,但两类拥塞控制机制仍有改进之处,特别是在网络状态检测方式、拥塞状态下的路由转发方面,有较高的研究价值。基于此,在本文中设计了一种综合型拥塞控制算法,算法分为网络拥塞检测和网络拥塞控制两个部分,对 NDN 拥塞控制机制加以改进。通过提前检测网络状态,在用户端和网内节点作出反应,达到缓解网络拥塞的目的。本章主要介绍综合型拥塞控制算法中拥塞检测部分的设计,通过对网络状态的检测,对不同程度的拥塞进行反馈,为后续根据不同的拥塞反馈采取的拥塞控制方案做准备。

3.1 综合型拥塞控制算法设计思路

网络拥塞被广泛认为是一个资源共享问题。在网络通信过程中,消费者发送请求到网络中,在等待请求内容返回时,所有消费者处于相同的网络带宽和缓存空间中,网络中路由节点的处理能力也是相同的。但是网络资源是有限的,并且在数据传输过程中,有可能突然出现用户需求猛增的情况。这种情况可能造成网络无法处理这些过量的需求,进而出现链路流量过载,传输效率下降的现象,最后导致网络拥塞。

在命名数据网络中,兴趣包的大小和数据包相比,兴趣包更小一些。因此数据包占据了网络流量的大部分,数据包的过量,用户的突发请求,网络资源不足等原因,可能导致网络发生拥塞。网络拥塞会对网络中数据的传输造成很大影响,例如出现数据传输往返时间增大、数据丢失、单位时间内传输数据的数量下降等问题。这些问题若不及时处理,网络传输性能出现恶性循环,用户端不断发送请求,下游链路不断向上游反馈通告,网络中产生大量冗余信息,网络链路持续过载,最终导致网络传输的崩溃。因此,高效解决网络拥塞问题是维持网络正常传输的关键。

拥塞问题的本质是存储转发节点无法处理网络中过量的数据分组。解决拥塞问题关键是确保网络能够承载用户提交的通信量,因此消除网络拥塞对用户来说意义重大。现存的 NDN 拥塞控制机制能够有效的利用网络资源,有效的对网络拥塞做出反应,但仍暴露出一些不足:

(1) 在网络状态检测方式上, 命名数据网络沿用传统的 TCP/IP 体系架构的检测方式, 总体分为三种, 超时重传 RTO 机制、利用缓冲队列以及综合数据往返时延和利用缓冲队列的方法。其中 ICP、CCTCP 等策略是超时重传 RTO 机制的代表, HoBHIS 等策略是后两者拥塞检测机制的代表。虽然目前的检测工作依赖这些策略, 但是这些策略在网络状态还未到达拥塞的情况下, 没有给出明确的网络状态结果。

(2) 在拥塞调节方面, 调整兴趣包发送速率是拥塞控制策略的核心。这种方式虽然思路清晰实现简单, 但是会因为盲目降低 Interest 数据包发送速率出现网络链路过于空闲, 不能根据不同路径之间网络状态的不同而差别对待, 降低网络整体的吞吐量和传输效率。

考虑到上面总结的问题, 我们认为对命名数据网络拥塞控制的研究, 既要考虑在用户端进行调节又要考虑中间节点的调节方案, 而 NDN 作为一种新型网络架构, 在数据传输方面和传统的 TCP/IP 体系架构有所不同, NDN 依据路由策略对用户请求进行逐跳转发, 从而有效调节网络拥塞状态。基于此, 本章提出了一种拥塞控制算法。如图 3-1 所示。算法设计思路主要采取以下两个方面: (1) 设计合理的拥塞检测方案对网络状态进行早期检测, (2) 在拥塞控制上, 采取双管齐下的策略, 一方面在用户端设计灵活的速率调节策略, 另一方面在网络内中间节点通过考虑拥塞状态的转发策略调节网络流量。因为这种算法综合了请求端和路由节点两方面进行拥塞控制, 所以我们称之为综合型拥塞控制算法 (Comprehensive Congestion Control Algorithm, CCA)。

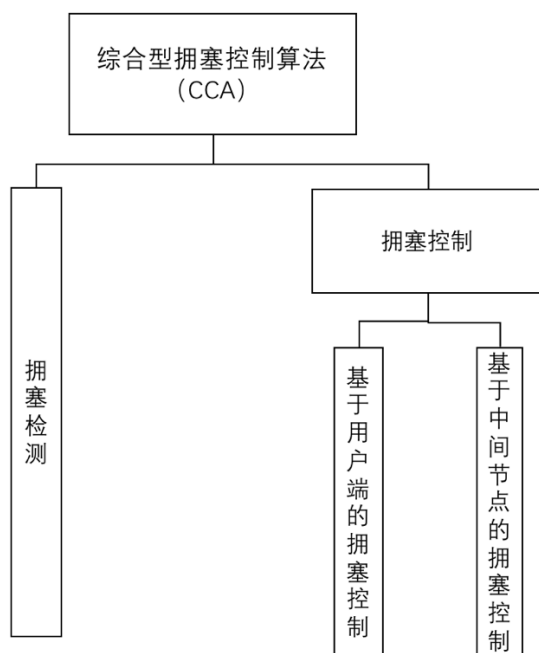


图 3-1 综合型拥塞控制算法

Figure 3-1 Comprehensive congestion control algorithm

3.2 拥塞检测设计

拥塞控制策略不仅要调节网络拥塞,还要采取一些措施避免拥塞情况的产生。对于网络链路中可能存在的情况,比如对已经发生拥塞的路径进行 NACK 通告,会引入更多负载,造成拥塞情况的加重。所以,进行网络拥塞检测时非常必要的,通过对网络状态的周期性检测,将检测结果以 NACK 消息通告的方式传输到上游节点,上游节点根据收到的反馈消息进行速率调节,从而确保网络传输的稳定和通畅。基于上述内容,本节将介绍 CCA 算法中设计的拥塞检测方案和具体的操作。

3.2.1 拥塞检测设计核心思想

在传统的 IP 网络中,通常采用计时器超时 RTO 机制来检测拥塞,超时产生的原因有很多,例如网络拥塞使得请求没有及时得到响应,另外网络链路出错也会引起超时。而这种机制无法区分是何种原因引起的超时,进而无法反应链路拥塞状态和链路拥塞的具体程度。NDN 网络作为新型网络架构,网络的中间节点不仅具备 IP 网络的转发数据功能,同时还可以进行一些复杂的操作,比如缓存数据内容和计算开销等。因此在中间节点上实时监测网络拥塞程度,将结果反馈给发送端,能更明显的减少反馈时间,并且在内容发送端明确网络状态,发送端能够及时处理。

在 NDN 网络中,一个兴趣包对应最多一个数据包,并且在内容体积上,兴趣包远小于数据包,进而可以忽略网络链路中兴趣包的流量,将网络拥塞产生的原因归结于传输路径中数据包的过量。同时,由于兴趣包和数据包之间的一对一数据流关系,所以可以通过统计兴趣包的数量推测即将返回的数据包的数量,进而推测网络链路的状态。

本策略的核心思想就是在网络中间节点实时统计某一条线路上兴趣包数量,将兴趣包数量作为判断网络状态的指标。然后根据不同的网络状态设置对应的 NACK,并将 NACK 信息反馈给上游节点。

3.2.2 网络状态的划分

在前文的介绍中我们通过统计链路上兴趣包的数量,来判断网络拥塞所达到的大致程度,并以此作为划分网络状态的指标。在真实网络中,网络拥塞等情况

的发生导致数据分组无法原路返回,进而出现丢包,因此一个兴趣包最多对应一个数据包。为了避免网络拥塞之后再采取措施,我们在指定网络拥塞标准时将标准值制定的越低越好,所以我们设定兴趣包和数据包是一一对应的,这样在推算数据包数量时会大于链路中数据包的实际数量,使得最后估算得出的网络拥塞程度比实际严重,进而能尽早采取措施,转移网络流量。

在 CCA 的拥塞检测机制中,我们在中间路由节点检测每个路由接口的瞬时兴趣包数量,最后计算一个周期内兴趣包的平均数,将此作为判断网络拥塞程度的指标。具体的说,网络检测将一个周期 T 均分成 i 段,每隔 T/i 个单位统计一次兴趣包的数量,一个周期一共计算了 $T/i+1$ 次。然后计算这个周期内兴趣包平均数 N_{avg} , N_{avg} 的计算方式见式(1-1):

$$N_{avg} = \frac{\sum_{i=1}^{T/i+1} N_i}{T/i + 1} \quad (3-1)$$

求出 N_{avg} 后,以此作为衡量网络拥塞的指标。然后根据不同的兴趣包数量将网络划分为三种状态区间,分别是网络空闲,网络繁忙和网络拥塞。拥塞程度依次递增,不同网络状态对应的兴趣包数量区间如表 3-1 所示:

表 3-1 不同的网络状态对应的兴趣包数量区间

Table 3-1 The number of interest packets corresponding to different network states

网络状态	兴趣包数量区间
网络空闲	$N_{avg} < N_{free}$
网络繁忙	$N_{free} \leq N_{avg} < N_{busy}$
网络拥塞	$N_{avg} \geq N_{busy}$

其中, N_{free} 是网络空闲和网络繁忙之间的临界值, N_{busy} 为网络繁忙和网络拥塞之间的临界值。当网络实际带宽达到设定带宽的一半时,此时我们在路由节点统计兴趣包数量,将计算得出的 N 作为 N_{free} 的值,当网络实际带宽与设定带宽相同时,我们将计算得出的 N 作为 N_{busy} 的值。因此,只需要将 T 时间内计算得出的 N_{avg} 与两个临界值进行比较,就可以得到当前的网络状态。

检测周期 T 的取值可根据当前的网络环境进行调整,当检测出的网络状态在两个相邻的状态区间之间切换或者一直处于一个网络状态时, T 的值调为初值的一倍,这样既能保证实时检测的原则,又可以降低由于网络检测带来的计算开销;当检测出的网络状态与上一次的状态不再相邻,例如拥塞检测的初始状态为空闲,由于某些原因,下一次执行拥塞检测后网络状态变为拥塞。出现这样的情况后, T 的值调为初始值的一半,增加拥塞检测频率,保证实时检测网络状态。

3.2.3 网络检测的结果反馈

确定当前的网络状态后,我们设计让路由节点向上游链路发送代表不同网络状态的 NACK 消息,通知上游链路当前的网络状态,以使上游节点能够及时调整接下来的兴趣包发送策略。

在 NDN 网络中,当一个路由节点中缓存的内容与兴趣包请求的内容不匹配,而且当前接口都无法转发兴趣包时,会通过 NACK 消息向上游链路通知可能发生的情况。在 NACK 结构中封装于此对应的兴趣包的信息和表示状态的错误代码。根据这些信息,网络链路可以很快接收到网络错误信息进而采取措施进行恢复。

NDN 目前设计支持三种 NACK 数据包,如表 3-2 所示: 1) NACK_LOOP: 收到兴趣包的路由节点,通过检测每个兴趣包唯一的 Nonce 值,判断是否收到了重复的兴趣包。当出现 Nonce 值相同的兴趣包时,会丢弃其他兴趣包只留下一个,并向上游节点通告出现环路的情况。2) NACK_GIVEUP_PIT: 路由节点收到兴趣包时,通过检查 FIB 表,判断是否能转发当前的兴趣包,当所有的接口都无法转发该兴趣包时,会返回此 NACK,让上游节点对兴趣包采取其他的转发策略。3) NACK_CONGESTION: 路由节点收到兴趣包时,通过检查路由接口发送兴趣包的速率,发现每一个接口都达到了速率限制,无法转发兴趣包,就会返回 NACK_CONGESTION 消息,通知上游节点发生了拥塞。上游节点收到 NACK 消息后,降低发送兴趣包的速率,缓解下游节点转发兴趣包的压力。上游节点收到不同 NACK 反馈的操作流程图如图 3-2 所示。

表 3-2 NACK 消息含义
Table 3-2 NACK message meaning

NACK 名字	含义
NACK_LOOP	下游链路出现环路
NACK_GIVEUP_PIT	下游链路无法转发兴趣包
NACK_CONGESTION	下游链路出现拥塞

由以上分析可知,只有当路由接口发送的兴趣包到达上限时,才会发送 NACK 通知拥塞,这种方式存在弊端: NACK 通告只反馈了拥塞的发生,并没有告知上游节点拥塞的具体程度,更不能区分是何种原因引起了拥塞;并且对网络状态的判断不够准确,由以上 NACK 消息种类的介绍可知,并没有一种 NACK 消息是描述下游节点还有可利用的空闲带宽,这就造成了空闲链路的浪费。

因此,本策略划分了三种网络状态,网络空闲、网络繁忙和网络拥塞。并针对这三种状态增加了三种不同类型的 NACK 消息,如表 3-1 所示。分别是 LESS_NACK 对应网络空闲状态, MORE_NACK 对应网络繁忙状态,

MOST_NACK 对应网络拥塞状态。通过反馈表示不同网络状态的 NACK 消息给上游节点，使上游节点能够根据 NACK 通告实施相应的速率调节策略，进而提高整个网络的链路利用率。

表 3-3 不同的网络状态对应不同的 NACK 消息

Table 3-3 Different network states correspond to different NACK messages

网络状态	NACK 消息
网络空闲	LESS_NACK
网络繁忙	MORE_NACK
网络拥塞	MOST_NACK

当执行 CCA 算法时，包含当前网络状态信息的 NACK 消息被路由节点反馈给上游链路，NACK 消息的反馈周期与拥塞检测周期一致。若检测到兴趣包数量小于 N_{free} ，则判定网络处于空闲状态，返回 LESS_NACK 消息；若检测到兴趣包数量介于 N_{free} 和 N_{busy} 之间，则判定网络处于繁忙状态，返回 MORE_NACK 消息；若检测到兴趣包数量大于 N_{busy} ，则判定网络处于拥塞状态，返回 MOST_NACK 消息。

3.3 本章小结

本章主要介绍的是 CCA 算法中的拥塞检测部分。对现有的拥塞检测机制加以改进，通过检测中间节点接口的兴趣包数量，作为指标来判断当前网络拥塞程度，并根据拥塞检测结果划分了三种网络状态：网络空闲、网络繁忙、网络拥塞，对应这三种网络状态相应设计了新的 NACK 消息，通过这些消息及时向上游链路进行反馈。通过这样的设计改进了 NDN 中网络状态反馈不及时，程度不够准确的问题，为后续的拥塞控制机制的实施做了准备工作。

第4章 综合型拥塞控制算法的速率调节设计

上一章介绍了 CCA 算法在拥塞检测方面的设计，这一章主要介绍当网络状态变为拥塞之后，CCA 算法在用户端和中间节点做出的速率调节设计。分别是在内容发送端设计灵活的速率调节方案，即基于接收端的速率控制。在中间节点通过改进转发策略的速率调节方案，即基于路由接口排名的转发策略。

4.1 基于接收端的速率控制

在 NDN 网络中，当路由节点收到下游链路反馈的拥塞通告时，由于 NDN 多路径路由的特性，会转发到除拥塞路径的其他路径中，当其他路径都不可用时，才会向上游的邻居节点通告。当网络拥塞一直没有得到疏通，每一个路由节点重复此过程，所有路径出现拥塞现象，没有可用路径转发兴趣包时，会生成一个 NACK_CONGESTION 消息，通过接收 NACK_CONGESTION 消息，路由器可以调节接口发送兴趣包的速率，从而使兴趣包转到其他接口转发。当网络持续不能满足当前需求时，NACK_CONGESTION 最终会反馈到内容接收端，也就是消费者节点，通常消费者节点会降低发送请求的速率，来解决网络拥塞问题。

在 IP 网络中，利用 TCP 协议进行拥塞控制，常采用 AIMD 算法进行速率调节，分为慢开始算法、拥塞避免算法、快速重传算法和快速恢复算法。还未发生拥塞时，发送窗口的增大规律表现为线性规律，也就是 AIMD 中的 AI (Additive Increase)，当网络中发生拥塞情况时，慢开始阈值改变，变为发送窗口的二分之一，即乘性缩小 MD (Multiplicative Decrease)。这种拥塞控制机制符合 IP 网络的架构设计，即端到端的通信设计。从拥塞检测，拥塞避免到拥塞控制，整个流程都是在通信传输的用户端发挥作用，从而达到缓解拥塞的目的。所以，基于接收端的速率控制采取 AIMD 算法进行调节，但由于 CCA 算法进行拥塞检测后，根据不同的网络状态需要采取不同的速率调节方案，因此我们可以改进 AIMD 算法，合理的进行速率调节。

对 AIMD 算法的修改思路如下。当收到 LESS_NACK 消息时，兴趣包发送速率是乘性增加 (Multiplicative Increase) 的，以便更好的利用空闲的网络链路；当反馈消息是 MORE_NACK 时，兴趣包发送速率是加性增加 (Additive Increase) 的，以线性规律增大拥塞窗口，在不发生拥塞的前提下，提高链路利用率；当反馈消息是 MOST_NACK 时，兴趣包发送速率以乘性规律减小 (Multiplicative Decrease)，尽快减小拥塞窗口。接收端收到扩展的 NACK 消息后的速率调整的过程图如图 4-1 所示。

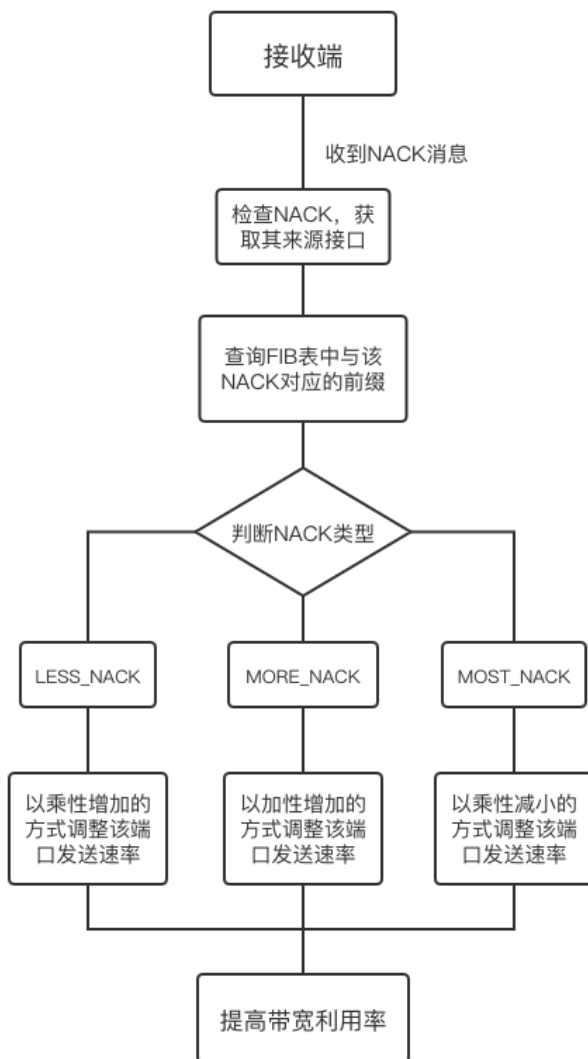


图 4-1 接收端速率调整过程

Figure 4-1 Receiver rate adjustment process

4.2 基于路由接口排名的转发策略

4.2.1 基于路由接口排名的转发策略的核心思想

本文 4.1 节介绍了 CCA 在接收端的拥塞控制方法：通过对 AIMD 算法的调整，实现了接收端的速率调节。本节将介绍 CCA 在中间节点的拥塞控制设计，我们称之为基于路由接口排名的转发策略：在转发层面对拥塞做出反应，通过对路由接口的内容命中率进行排名，将兴趣包尽可能从命中率高的接口转发出去，降低无用接口的转发率，从而提高整个网络的传输效率。

在介绍该策略之前，我们首先看一下 NDN 的转发过程。Interest 数据包和 Data 数据包的转发过程如图 4-2 所示。假设用户端请求名字前缀为

/pics/picture.png 的文件。由于文件大小过大，内容源节点通常会将一个文件分成多个小的数据包来返回。用户端的目的是得到一个完整的文件，此时用户请求端会多发送几个兴趣包。当名称为/pics/picture.png/s0 的兴趣包根据转发策略转发到某一路由节点时，该路由节点会先通过名字前缀查找内容缓存表(Content Store)，若在 CS 表中找到兴趣包请求的内容，路由节点会将该内容返回给用户端。当名字前缀为/pics/picture.png/s1 的兴趣包到达该路由节点时，没有在 CS 表中查询到请求的内容，路由节点就会将该名称写入 PIT (Pending Interest) 表中,更新该兴趣包到达的端口号。最后，路由节点使用最长前缀匹配策略查找 FIB (Forwarding Information Base) 表，找出能够转发该兴趣包的接口，并根据转发策略将兴趣包通过这些接口转发出去。以上就是路由节点转发兴趣包的过程。

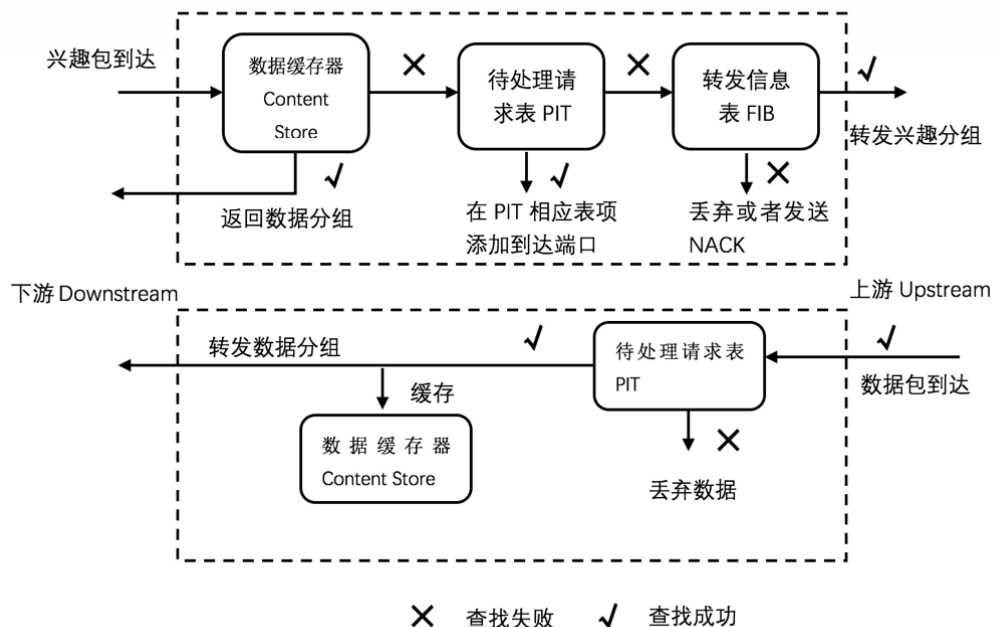


图 4-2 NDN 数据转发过程

Figure 4-2 NDN data forwarding process

在 NDN 的转发过程中，当兴趣包被转发到包含用户所需内容的路由器时，兴趣包命中，路由节点将请求内容沿兴趣包转发的路径原路返回到用户端。在这个过程中，通过名字前缀识别兴趣包，通过路由节点中记录的状态信息转发数据包。兴趣包和数据包不记录通信两端的信息。当出现多个请求重复内容的兴趣包时，路由节点会将这些兴趣包的信息更新到 PIT 表中，然后转发第一个兴趣包到网络中。当 Data 数据包到达时，路由器会查询 PIT 表中兴趣包的信息，然后通过兴趣包的进接口依次将数据包转发出去。最后，路由器更新 PIT 表删除记录并且更新 CS 表将数据包缓存。

由上述转发过程可知，FIB（转发信息表）与网络状态密切相关。在传统的 TCP/IP 体系架构中，FIB 表存放的是最优下一跳的路由地址，NDN 与之不同，NDN 网络的 FIB 表项中存放一个包含多个接口的列表，而且这个接口列表是经过排序的。FIB 表结构如图 4-3 所示。每一条记录包含多个字段，分别是名字前缀、过期时间和多个经过排序的转发接口。对于一个名字前缀的每一个转发接口，FIB 中保存了这个接口的 ID、路由优先级、往返时间、接口状态以及速率限制信息。

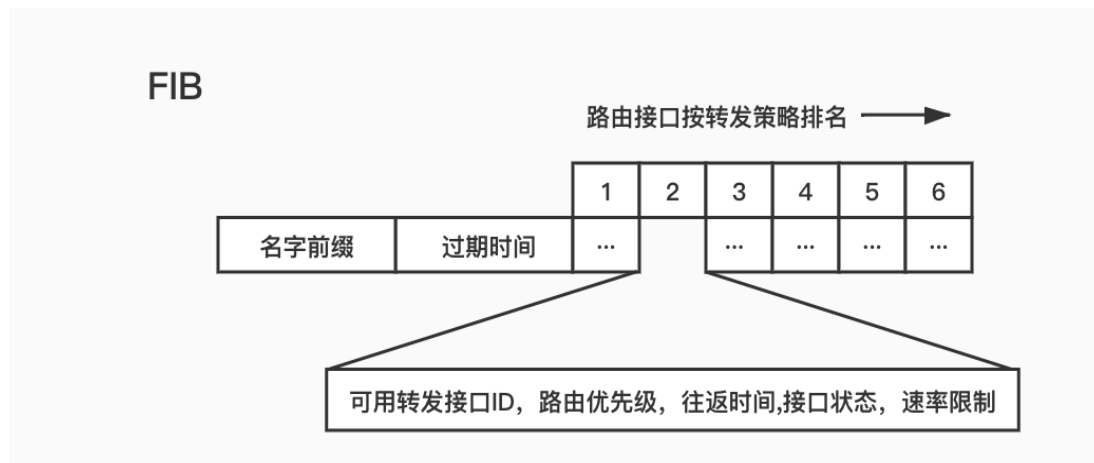


图 4-3 FIB 数据结构

Figure 4-3 FIB data structure

在 NDN 网络的转发过程中，当路由节点识别到达的兴趣包没有对应的路由转发信息时，会先根据 FIB 中的路由优先级字段对接口进行排序；直到获取到转发层的性能信息，网络中的转发策略才会综合路由优先级和转发层的性能这两种信息，决定接口的排序。而本转发策略的核心思想是提出了命中率优先级（hit-rate preference）的概念，并以此决定路由接口的排序。命中率优先级是路由接口转发的某一名字前缀的兴趣包在该接口发生命中的概率，具体的计算方式在 4.2.4 小节给出。命中率优先级与 NDN 中路由优先级字段的不同之处在于，命中率优先级概念依赖 CCA 算法的执行。整个过程为对网络进行拥塞检测后，获取到当前的网络状态，进而实时更新命中率优先级，选择命中率更高的路由接口进行转发。

4.2.2 数据结构分析

根据之前的思路，我们需要在路由器中构建一个命中率优先级表（Hit-rate Preference Table）用来记录每个路由节点所有接口转发且命中的概率，以此概率来排序，并为这个记录设置一个生命周期。

HPT 的具体记录如 4-1 表所示：

表 4-1 HPT 数据结构
Table 4-1 HPT data structure

字段名	含义
name	接口名
hit-rate preference	命中率优先级
ttl	记录的生存时间, 当时间消耗为 0

上表中, name 字段用来记录节点的接口名, hit-rate preference 记录每个接口的命中率优先级, ttl 表示该接口的命中率优先级的生命周期, 除了该接口发生命中, 更新命中率优先级以外, 针对每个接口, 以 ttl 为周期更新该接口的命中率优先级。

4.2.3 基于路由接口排名的转发策略具体流程

根据 NDN 的运行机制, 一个 Interest 数据包可以取回最多一个 Data 数据包, 当本地缓存无法命中时, 经过某个接口转发出去的兴趣包不一定能返回对应的数据包, 此时不仅内容请求方得不到满足, 还有可能在已经拥塞的链路上引入更多的负载, 加剧网络拥塞程度。因此, 尽可能让兴趣包通过命中率高的接口转发出去, 减轻网络拥塞程度, 提高内容命中率。CCA 算法在路由节点部署后执行, 具体过程为, 首先通过计算路由接口的内容命中率构建和维护 HPT; 其次当兴趣包到达路由节点时, 查询 HPT 记录的路由接口排序, 决定兴趣包往哪些接口转发。

假设在观测时间 T 内路由器 R_i 的第 j 个接口接收数据包的数量为 S_j , R_i 有 N 个可用的接口, R_i 的 FIB 表中的有一条记录为名字前缀 prefix, j 接口返回的匹配 prefix 的数据包数量为 S_{match} , 则 j 接口的命中率为

$$H_j = \frac{S_{match}}{\min(S_j, \frac{S_j}{\sum_{j=1}^N S_j} \times Max \times T)} \quad (4-1)$$

其中, Max 表示该路由节点 R_i 能接收的最大数据包数量; $\frac{S_j}{\sum_{j=1}^N S_j} \times Max \times T$ 表示 T 时间段内接口 j 能接收的最大数据包数量。 $\min(S_j, \frac{S_j}{\sum_{j=1}^N S_j} \times Max \times T)$ 表示在接口 j 实际接收到的数据包数量和接口 j 能接收的数据包最大值之间取较小的值, 也就是说接口 j 接收到的数据包数量最大不超过它的极限值。 H_j 越大, 表示接口 j 转发前缀为 prefix 的兴趣包命中率越高。

在 HPT 表中包含的是接口的命中率优先级, 是转发策略依据的决定路由接口排名的字段, 代表的是在整个路由节点所有接口命中率的占比, 所以接口 j 的命中率优先级定义为

$$H_{forward} = \frac{H_j}{\sum_{j=1}^N H_j} \quad (4-2)$$

其中, $\sum_{j=1}^N H_j$ 表示路由节点 R_i 中 N 个接口的数据包命中率和, 这里假设该路由节点所有接口都可以转发用户请求的前缀为 **prefix** 的兴趣包。

本策略的实施包括两个步骤: 初始化 HPT 表与运行 HPT 表。前一步骤用于评估路由节点各转发接口在初始阶段的命中率, 后一步骤根据节点接口的命中率设置命中率优先级, 并对命中率优先级进行动态更新。

初始化 HPT 表阶段, 将路由节点各接口的命中率默认设置为 1, 在该步骤中: ①若前缀为 **prefix** 的兴趣包在路由节点 R_i 的缓存中命中, 则直接返回内容数据包; ②若前缀为 **prefix** 的兴趣包在路由节点 R_i 未命中, 路由节点以相同概率选择接口转发该兴趣包, 当转发出去的兴趣包在其他节点命中, 数据包通过原接口返回到节点 R_i , 此时统计出该接口到达的总数据包数量, 计算出接口命中率 H_j 和命中率优先级 $H_{forward}$, 写入 HPT 表。

运行 HPT 表阶段, 若前缀为 **prefix** 的兴趣包在路由节点 R_i 缓存中未命中, 节点 R_i 查询 HPT 表中的命中率优先级 $H_{forward}$, 根据此字段决定的路由接口排名进行转发。当转发出去的兴趣包在其他节点命中, 数据包通过原接口返回到节点 R_i , 节点同样再次计算该接口的命中率优先级 $H_{forward}$, 以时间 **ttl** 为周期更新 HPT 表。

本策略的流程图如图 4-4 所示。

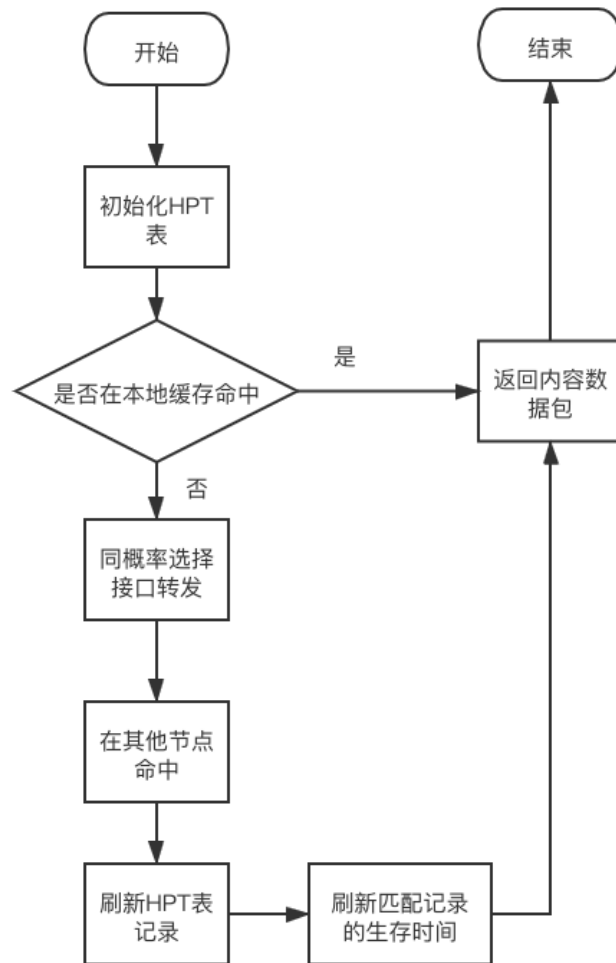


图 4-4 算法流程图

Figure 4-4 Algorithm flow chart

4.3 本章小结

本章主要介绍的是针对网络在拥塞状态下用户端和中间节点的拥塞控制策略。基于网络状态拥塞这一场景，在用户端采用灵活的 AIMD 算法，根据接收到的不同警告程度的 NACK 消息进行不同的速率调节；在网络中间节点通过调节路由节点的转发接口的排名顺序，使兴趣包尽可能的通过命中率高的路由接口转发出去。由转发策略具体流程设计的描述可见，本策略实现了较为客观的接口命中率评估，以大概率选择命中率高得接口将用户请求转发至可靠的内容源，在降低网络拥塞程度的同时，周期性的更新 HPT 表，实时更新不同的内容源路径，动态调节网络状态。

第 5 章 算法的实验与分析

前文中对 CCA 的拥塞检测机制，基于接收端和中间节点的拥塞控制机制进行了详细的描述。拥塞检测机制主动检测兴趣包情况获取网络状态，描述网络具体的拥塞程度，两种拥塞控制策略分别从接收端调节兴趣包发送速率和从中间节点提高兴趣包命中率两个方面实施拥塞控制，在本章中会将算法放在 ndnSIM 平台上进行实验模拟，并对实验结果进行分析论证。

本章首先介绍实验的模拟环境，简单介绍 ndnSIM 平台，然后对整体实验进行设计，包括具体的实验场景以及算法的实现，设定实验的评估指标，最后对实验数据进行分析。

5.1 ndnSIM 模拟平台

ndnSIM 实验平台的目的是模拟 NDN 的协议。ndnSIM 模拟实验平台是 NS-3 的一个模块，NS-3 是一个网络模拟器，它可以支持 Linux 系统，NS-3 是 NDN 研究者们常用的网络仿真环境，可以进行 IP 或者非 IP 网络的仿真研究。在 NS-3 中，ndnSIM 是以一个新的网络层协议模块存在并实现的。

根据 NS-3 其他模拟实验的设计规则，ndnSIM 的设计理念为模块化设计，它实现了一个新的基于网络层的模型，ndnSIM 的优点是，当对 NDN 进行新的算法设计时，例如路由转发、缓存策略、拥塞控制等，可以在这些算法所在的包内进行改动，不会污染全局组件。ndnSIM 平台的架构图如图 5-1 所示，从图中可以看到 ndnSIM 中包含了 NDN 网络中的数据结构，包括 PIT 表、CS 表、FIB 表、路由器中的接口和路由转发策略等。平台通过 C++ 语言实现，可以修改默认的缓存与转发策略，扩展缓存与转发策略的功能，编写实现研究人员提出的新的算法并进行实验。

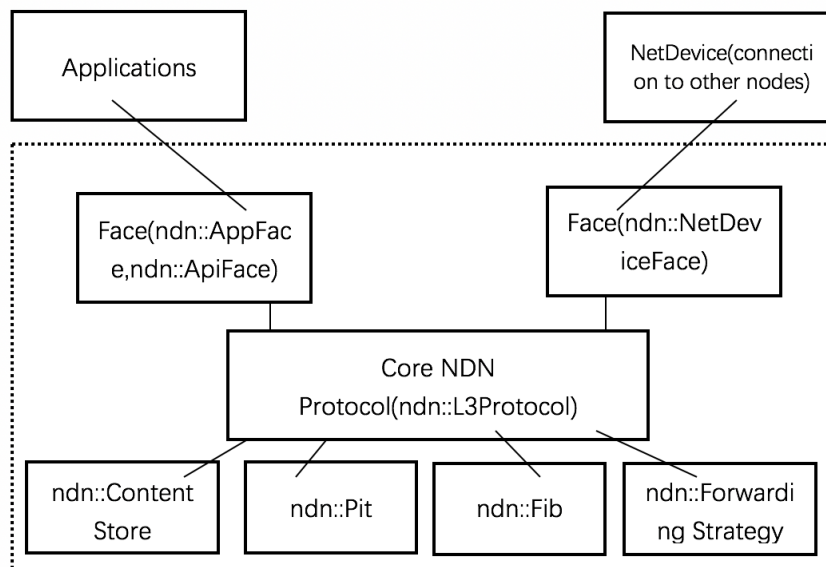


图 5-1 ndnSIM 架构图

Figure 5-1 ndnSIM architecture diagram

ndnSIM 平台由不同的模块组成，这些模块各自封装了相应的功能，彼此之间紧密联系又保持各自的独立。其中一些关键的模块在模拟实验中发挥了重要作用。ndn::L3Protocol 实现了 NDN 网络中的各项协议，例如转发兴趣包需要的路由协议等，通过 ndn::L3Protocol 在路由节点安装 NDN 协议，实现网络的通信传输。NFD 模块是 ndnSIM 平台的核心模块，它提供了 NDN 网络的正常通信的功能，即兴趣包和数据包的转发能力。NFD 模块包含 nfd::Forwarder、nfd::fw::Strategy、nfd::Face、和数据传输过程中三种重要的数据结构 nfd::Fib、nfd::Pit、nfd::ContentStore。nfd::Forwarder 是 NFD 中的主要功能模块，它管理着路由节点中的数据结构。在 nfd::fw::Strategy 的 fw 文件夹中存放着各种转发策略，实现本文的转发策略就需要修改 fw 中的 forwarder.cpp 文件。而 NDN 兴趣包和数据包转发过程中需要的路由接口是在 nfd::Face 模块实现的。在 NFD 模块中，需要多个工作流程才能完成对 NDN 中数据包的转发，如图 5-2 所示，

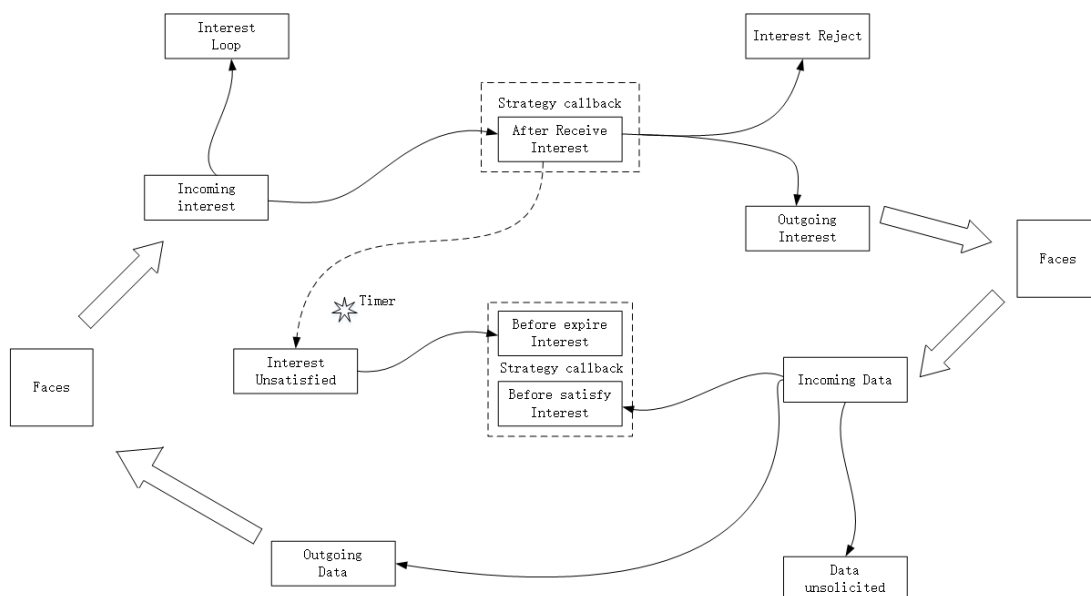


图 5-2 ndnSIM/NFD 工作流程概况

Figure 5-2 Overview of ndnSIM / NFD workflow

在兴趣包的转发过程中，需要经过兴趣包流程、超时流程和数据包流程才能完成 NFD 模块的转发工作。具体来说，NFD 需要对兴趣包的转发时机、转发路径和是否转发做出决策。NFD 模块不仅为 NDN 网络的正常传输提供了基础，同时还为研究人员扩展 NDN 网络功能提供了空间。

5.2 模拟实验基本设计

上一小节介绍了 ndnSIM 平台，本节将在 ndnSIM 平台上进行模拟实验。本文使用 Ubuntu 16.04 版本和 ndnSIM2.3 版本进行实验。网络拓扑采用树形结构，如图 5-3 所示。

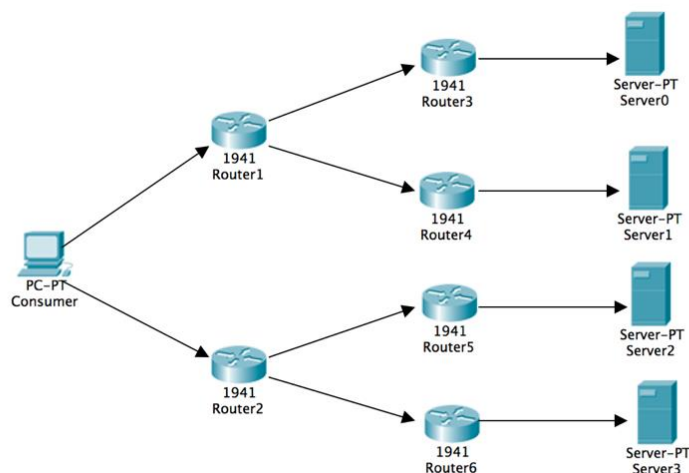


图 5-3 树形网络拓扑

Figure 5-3 Tree network topology

Consumer 为网络中的消费者节点, Router1、Router2、Router3、Router4、Router5、Router6 为网络中的路由节点, Server0、Server1、Server2、Server3 表示网络中的生产者节点。每两个相邻的路由节点之间为一跳, 每一跳之间的链路延迟为 10 毫秒 (ms), 链路带宽设置为 10Mbps。兴趣包的大小为 40Bytes, 数据包的大小为 2000Bytes。兴趣包的发送频率为 50 个/秒, 仿真实验的运行时间为 100s。在同一个网络拓扑结构下, 本实验与 AIMD 算法和最佳路由策略 (ndn::fw::BestRoute) 来进行对比, 使用 ndn::L3RateTracer 生成 trace 文件, 导入 R 语言软件中生成数据图, 对比分析数据得出实验结论。

5.3 算法的模拟实验

本实验的算法需要修改 ndnSIM 中的转发模块。主要集中在路径 NFD/daemon/fw 下的 forwarder.cpp 文件中。同时, 还需要在其他模块新建类和函数, 例如扩展 NACK 消息、修改本实验用到的 NDN 中的基础数据结构。另外还需要考虑在 ndnSIM 平台的合适位置执行本文提出的 CCA 算法。下面将简述本文在实验中对 ndnSIM 做出的修改。

首先修改 NDN 网络中的基础数据结构。根据 CCA 算法中拥塞检测机制的设计, 当检测完网络状态后, 需要发送相应的 NACK 包进行反馈, 在 NDN 中, 一个 NACK 和其对应的兴趣包遵循同一个名字和 Nonce 值, 除此之外, NACK 消息还封装了一个错误码, 这个错误码描述了此 NACK 的含义。本实验中需要向 NACK 包中添加类型标识字段, 用于标识不同的 NACK 包。具体需要修改 /ndn-cxx/src/lp 目录下的 tags.hpp、tlv.hpp 和 field.hpp 三个文件, 这些文件包含了

关于包传输的一些定义。本实验中，向 `tags.hpp` 文件中添加了 `TagType` 标记，用于记录 NACK 的类型，设置为 3 个值，0 对应 `LESS_NACK`，1 对应 `MORE_NACK`，2 对应 `MOST_NACK`，默认为 2。在生产 NACK 包的地方和转发 NACK 包的地方插入标签 `interest->setTag(make_shared<lp::TagType())`，从而完成对 NACK 包中添加字段的操作。

然后构建基于路由排名的转发策略所需要的命中率优先级表 HPT。通过在路由器中构建的 HPT 表来记录路由接口优先级排名，在路径 `NFD/daemon/table` 下建立新目录 `rt`，并在此目录下建立 `hpt.hpp` 和 `hpt.cpp` 两个文件，这两个文件用来构造 HPT 表的数据结构。该数据结构通过链表结构来记录路由接口排名的实时状态。

在 HPT 表中，`createRecord(face)` 方法用于创建记录、`deleteRecord(face)` 方法用于删除记录、`refresh()` 方法作为刷新 HPT 表中接口排名的生命周期。然后添加到 `NFD/daemon/fw` 目录下的 `forwarder.hpp` 文件中。通过调用 HPT 中的方法来查询路由接口和转发兴趣包。

最后在 `examples` 目录下的 `ndn-tree-delay-tracer.cpp` 文件中运行主程序。通过 `ndn::StackHelper` 构建路由节点，然后设置缓存策略。通过 `ndn::AppHelper` 构建消费者与生产者，并安装路由节点，由 `ndnHelper.setCsSize(n)` 语句确定路由节点的内存。在对比实验中启用最佳转发策略。最后最为关键的是生成 `tracers` 文件，`tracer` 是 `ndnSIM` 提供的用于记录数据的文件，其中有三个使用率较高，包括了一些很关键的性能指标，比如网络流量大小、网络延迟时间、用户请求的路径长度。这三个文件分别是 `L3RateTracer`、`CsTracer` 和 `AppDelayTracer`。如图 5-4 所示。

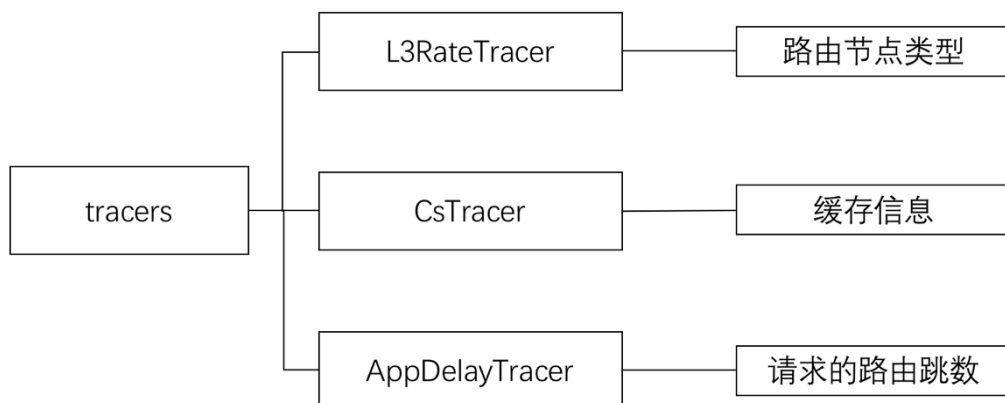


图 5-4 tracers 文件种类
Figure 5-4 Types of tracers files

其中 NDN 网络中路由节点的种类存放在 L3RateTracer 文件中, 具体包括兴趣包经过路由节点时留下的信息, 分别是 InData 对应兴趣包到达节点的信息, OutData 对应兴趣包通过路由接口转发出去的信息, 除此之外, L3RateTracer 文件还存有一些其他的内部报文种类, 该文件在最后一列的 KilobytesRaw 字段下记录了网络流量大小这一性能指标。关于路由节点的缓存信息则存放在 CsTracer 文件中, 具体包括路由节点缓存命中的情况 (CacheHit) 和缓存未命中 (CacheMisses) 的情况, 统计该文件的信息可以评估路由节点的缓存命中性能。关于内容请求的路由跳数信息保存在 AppDelayTracer 文件中, 该文件的 HopCount 字段记录网络中所有节点的路由跳数, 通过这个字段可以得出路径长度, 而 DelayS 字段记录了网络延迟时间, 进而可以推算出整个网络的延迟时间。本实验用 L3RateTracer 文件记录 ndnSIM 模拟环境产生的日志, 日志中包含了网络中各个路由节点在不同时间执行任务的信息, 在 Simulator::Run() 语句之前添加 ndn::L3RateTracer::InstallAll("lxy/tracers-num.txt", Seconds(0.5)) 语句, 生成如图 5-5 所示的文件, 将产生的实验数据放入 R 语言软件进行处理, 进而生成数据图表。

Time	Node	FaceId	FaceDescr	Type	Packets	Kilobytes	PacketRaw	KilobytesRaw
0.5	0	1	internal://	InInterests	0	0	0	0
0.5	0	1	internal://	OutInterests	0	0	0	0
0.5	0	1	internal://	InData	0	0	0	0
0.5	0	1	internal://	OutData	0	0	0	0
0.5	0	1	internal://	InNacks	0	0	0	0
0.5	0	1	internal://	OutNacks	0	0	0	0
0.5	0	1	internal://	InSatisfiedInterests	0	0	0	0
0.5	0	1	internal://	InTimedOutInterests	0	0	0	0
0.5	0	1	internal://	OutSatisfiedInterests	3.2	0	2	0
0.5	0	1	internal://	OutTimedOutInterests	0	0	0	0

图 5-5 L3RateTracer 文件中的部分数据

Figure 5-5 Part of the data in the L3RateTracer file

5.4 实验评估性能指标

本次实验在三个方面与 ndnSIM 默认的 AIMD 算法和最佳路由策略进行对比, 这三个方面分别是链路带宽利用率、路由节点内容命中率、用户总命中率。通过这三个性能指标能够综合全面的反应本文的 CCA 算法的优劣。

链路带宽利用率: 链路带宽利用率直观的反映了拥塞控制策略的性能。当用户发送内容请求到网络中时, 我们期望网络传输链路能得到较好的利用, 减少网络中出现丢包和空闲的情况。本文的链路带宽利用率计算方法是当前链路的数据流量和链路带宽的比值。

路由节点内容命中率: 表示用户请求在该路由节点的接口被命中的概率, 是指当本地缓存没有命中, 通过路由节点接口转发出去的兴趣包被命中的概率。这个指标比较直观的反应了基于路由接口排名的路由转发策略的有效程度, 该指标的计算方法和前文第四章中的接口命中率 H_j 相同。

用户总命中率：表示用户将请求发送到整个拓扑网络中，经过路由节点的转发后，请求被满足的概率。这个指标在宏观上反应了经过 CCA 之后，在用户请求方面的改善程度。该指标的计算方法是网络中命中的数据包总量与用户发送兴趣包总量的比值。

5.5 实验结果

本文提出的 CCA 算法最突出的贡献之一就是提高了兴趣包命中率，在转发层面缓解了网络拥塞，提高了网络传输效率。本次模拟在请求相同数据量的情况下，对比了 CCA 算法和 NDN 默认策略之间的性能差异，并且在模拟 CCA 算法时，采用控制变量的对比实验，对 CCA 的性能进行分析。根据分析得出结论，CCA 算法在很大程度上的利用了链路带宽，大大降低了链路传输的开销，在路由转发缓解网络拥塞状态方面，CCA 有巨大的优势所在。

按照前文陈述的实验设计与实现过程，对实验得出的数据进行对比分析。除了和 NDN 默认策略进行对比之外，作为对照，本次实验选取在用户端请求 5M 和 10M 数据作为对比进行验证。

如图 5-6 的本文的 CCA 算法和 NDN 默认的 AIMD 算法对比图所示，图 5-6 显示的是 Router1 到 Router3 之间的带宽利用率，用户向 Server 请求 5M 的数据，由图中可以看出，由于网络状态检测策略的实施，本文的拥塞控制策略更快的使用了链路，因此更早的探测到网络拥塞的发生，更早做出反应。说明本文算法在拥塞预测方面具有更高的灵敏性。

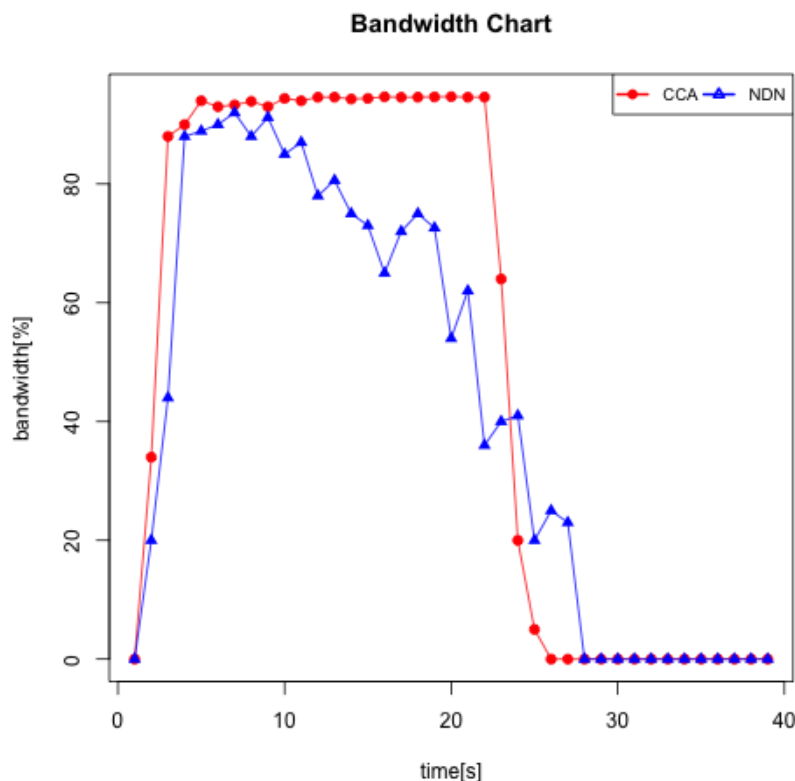


图 5-6 请求 5M 内容数据路径 Router1、Router3 的链路带宽利用率

Figure 5-6 Requesting link bandwidth utilization of 5M content data paths Router1 and Router3

另外,在拥塞控制方面,和 NDN 采用的 AIMD 算法相比,本算法通过拥塞检测获取网络不同的拥塞程度,并针对不同的拥塞程度采用不同的速率调整方案。在图 5-6 中可以看出,CCA 的速率调节是优于 AIMD 算法速率调节方案的,CCA 在请求数据时,带宽利用率基本维持在 90% 以上,而 AIMD 算法在 22s 左右带宽利用率一直在 50% 以下,带宽利用率一直反复波动,体现了 AIMD 算法的弊端,网络空闲时没有较快占据带宽,链路流量增大到一定程度后,请求超时,拥塞窗口减半。而且请求 5M 的数据,CCA 比 AIMD 算法少用 3s,传输效率更高。综上分析说明网络状态拥塞时 CCA 算法在调节网络状态和稳定传输方面都优于 AIMD 算法。

如图 5-7 所示,相同条件下本文的基于路由接口排名的转发策略和 NDN 最佳路由转发策略在路由节点内容命中率方面的性能对比图,图 5-7 显示的是拓扑结构中的 Router1 的内容命中率。由图中可以看出,在初始化阶段,两种策略的命中率是相同的,随着时间的推移,由于 CCA 不断更新接口命中率优先级,使兴趣包通过排名靠前的接口转发出去,实施 CCA 的路由器命中率不断升高进而趋于稳定,而最佳转发策略始终没有达到 CCA 的内容命中率,并且有下降趋势。随着时间的增加,网络中流量增大,在 CCA 算法下,节点 Router1 选择命中率

高的接口转发兴趣包的概率增大,因此用户请求产生的网络流量更大概率被转发到内容源节点,从而降低了网络流量升高对 Router1 内容命中率产生的影响。

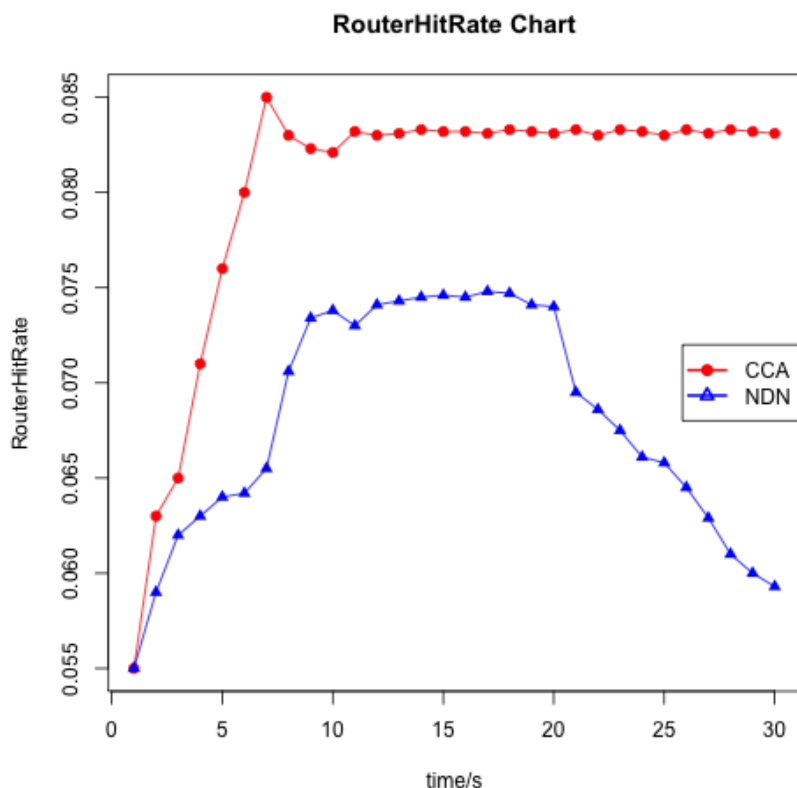


图 5-7 请求 5M 内容数据 Router1 的内容命中率

Figure 5-7 Content hit rate of Router1 requesting 5M content data

如图 5-8 所示,显示的是拓扑结构中的 Router3 的内容命中率。由于 Router3 距离内容提供者 Server0 较近,可以看出,Router3 在两种策略下的内容命中率都是高于 Router1 的。在图 5-8 中可以看出,随着网络流量的增加,Router3 的内容命中率有小幅波动,但整体趋势是稳定的。但最佳转发策略中,内容命中率波动较大,在峰值维持时间不久就呈现下降趋势。由于网络流量的增大,内容命中率与 CCA 相比差距增大,由此可见,CCA 在链路拥塞的情况下内容命中率和稳定性都比最佳转发策略优秀。

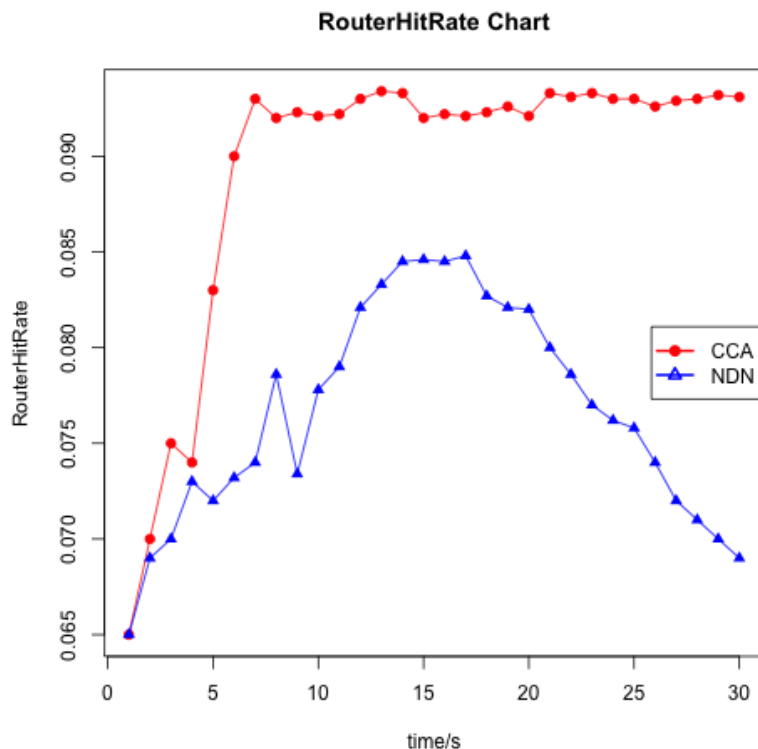


图 5-8 5M 内容数据 Router3 的内容命中率

Figure 5-8 Content hit rate of 5M content data Router3

如图 5-9 所示, 由于绘制用户总命中率没有明显差距, 因此用 100% 减去用户总命中率, 得到用户总丢包率, 由此绘制的数据图更直观明显的反应了 CCA 算法和 NDN 默认策略的差别。图 5-9 中, 分别在算法运行的不同时间统计了三组数据, 结果显示, 本文提出的拥塞控制算法在用户总命中率方面都是优于 NDN 默认策略的。

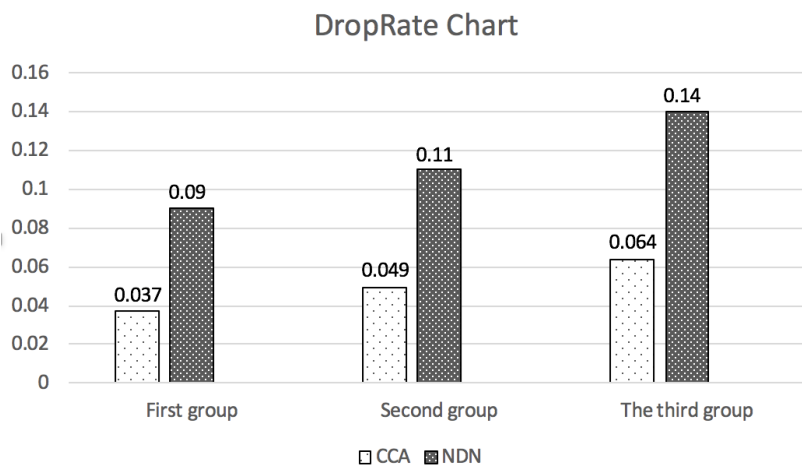


图 5-9 5M 内容数据用户总丢包率

Figure 5-9 Total packet loss rate of 5M content data users

如图 5-10 所示, 用户向 Server 请求 10M 数据的 Router1 到 Router3 之间的带宽利用率, 由图中可以看出, 本文的拥塞控制策略更早的占据了链路带宽, 并更早的完成了数据传输, 请求相同 10M 数据 CCA 在 26s 完成, NDN 的 AIMD 算法则用了 38s, CCA 减少了 31.5% 的传输时间。随着时间推移, AIMD 算法带宽利用率出现了较大起伏, 这个现象说明当网络流量增加时, NDN 中 AIMD 算法的拥塞控制和速率调节发挥不出应有的作用。

相比于请求 5M 数据的实验结果对比来看, 请求数据量增大一倍, 从图 5-10 可以看出, CCA 的传输时间较图 5-8 没有较大增长, 而 AIMD 算法的传输时间则为上一个实验的 1.36 倍, 相比于 AIMD 算法, CCA 的传输效率更高。

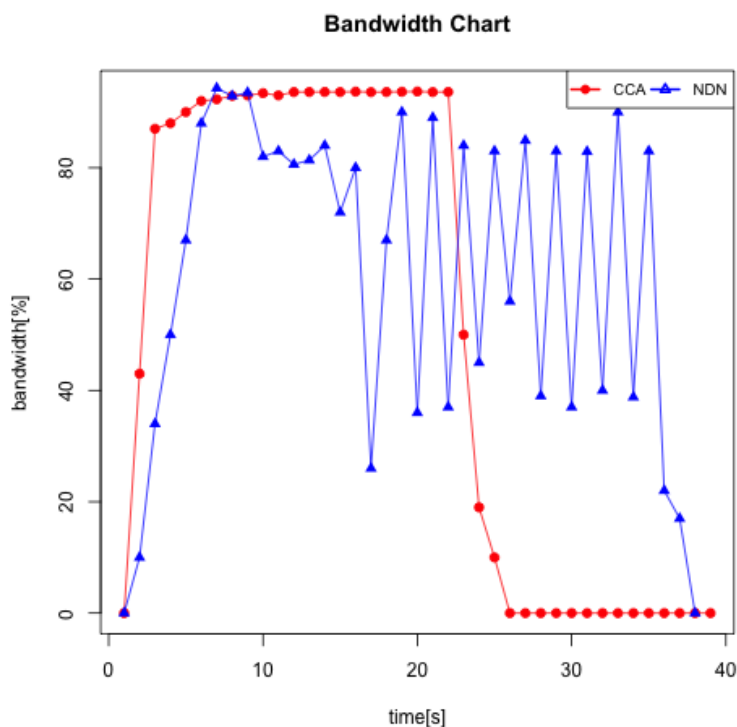


图 5-10 10M 内容数据路径 Router1、Router3 的链路带宽利用率
Figure 5-10 10M content data path Router1, Router3 link bandwidth utilization

如图 5-11 所示, 将数据量增加到 10M 之后路由器 Router1 的内容命中率, 由图中可以看到, CCA 算法下的 R1 内容命中率先快速上升后整体趋于稳定, 而 NDN 的最佳转发策略下的 R1 内容命中率始终没有达到 CCA 的内容命中率, 并且有下降趋势。对比数据内容为 5M 的实验, 数据量增加后, 在峰值 CCA 的命中率下降了 24%, 最佳转发策略下降了 43%。如图 5-12 所示, 在峰值 CCA 的命中率相较于图 5-7 下降了 21%, 最佳转发策略下降了 46%。由此可见, 随着网络中流量增加, 本算法在内容命中方面的性能优于 NDN 的最佳转发策略。

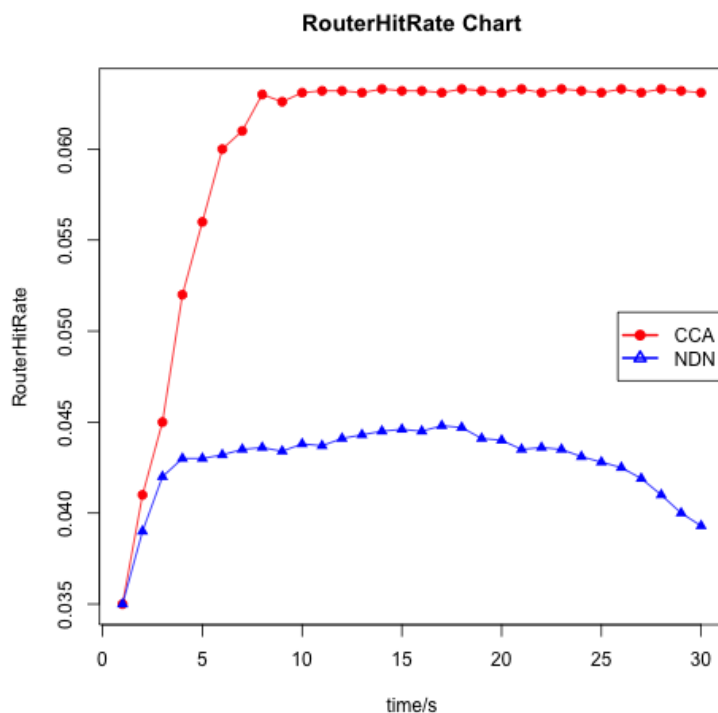


图 5-11 10M 内容数据路径 Router1 的内容命中率
Figure 5-11 10M content data path Router1 content hit rate

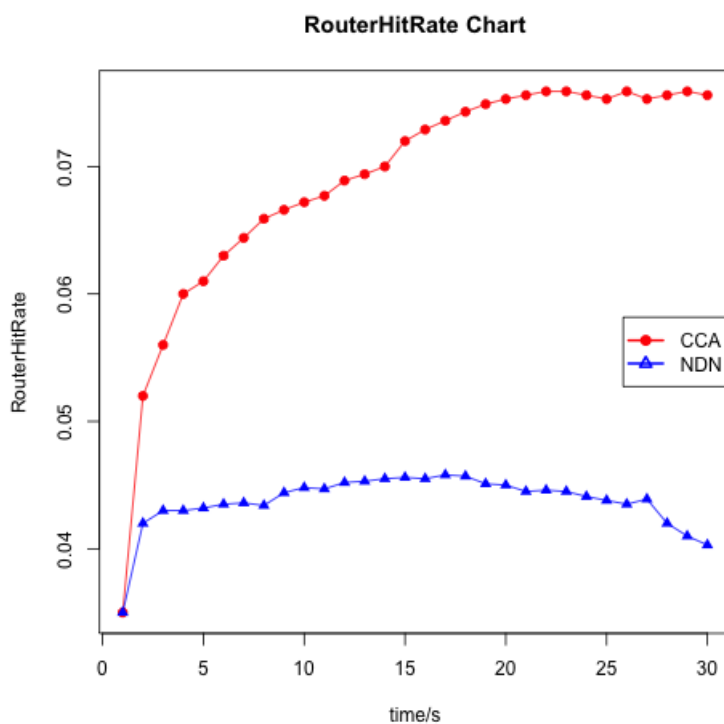


图 5-12 10M 内容数据 Router3 的内容命中率
Figure 5-12 Content hit ratio of 10M content data Router3

如图 5-13 所示, 在数据量为 10M 的实验中同样统计三组数据, 由图中可以看出, 每一组 CCA 的丢包率都是小于 NDN 默认策略的。相较于请求 5M 数据内

容的实验，两种策略的丢包率都有所上升。综上说明，本算法在请求数据量的增大的条件下性能还有待提升。

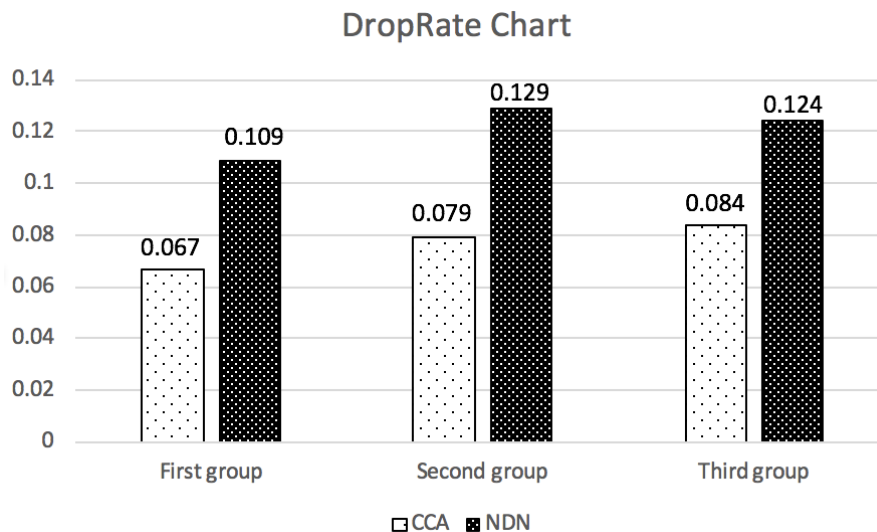


图 5-13 10M 内容数据用户总丢包率

Figure 5-13 Total packet loss rate of 10M content data users

5.6 本章小结

本章主要在 ndnSIM 模拟平台下，将本文 CCA 算法设计的拥塞检测机制，基于接收端的速率调节和基于路由接口排名的转发策略，与 ndnSIM 平台中已有的拥塞控制策略和最佳转发策略进行实验数据的对比，对比的性能指标从链路带宽利用率、路由节点内容命中率、用户总命中率三个方面进行比较。可以发现本算法在这些性能指标方面具备很多优势，在拥塞控制方面，CCA 实现了实时检测网络状态，并根据不同的网络状态在接收端调整网络流量，在中间节点提高内容命中率，达到提高网络传输效率，缓解拥塞的目的。

结 论

命名数据网络是 21 世纪新兴的未来互联网体系架构, 研究前景广阔, 是未来互联网变革性的体现, 对于 IP 网络出现的不足和弊端, NDN 提出了革命性的设计思维, 为研究人员提供了探索未来网络的研究思路。在用户规模日益剧增的环境下, NDN 的出现在学术界和产业界掀起了研究热潮。近些年, 国内外研究学者对 NDN 的研究逐步加深, 在命名机制、拥塞控制、缓存策略、转发策略等方面都颇有造诣, 为今后的研究工作打下了坚实的理论基础和实际经验。

本论文的研究课题是命名数据网络中的拥塞控制策略, 主要是根据目前主流的 NDN 网络拥塞调节机制, 针对用户端和逐跳式两个方面, 设计出高效的速率调节算法, 在对网络进行拥塞检测之后, 从接收端和中间路由节点作出响应, 灵活调节兴趣包发送速率, 提高兴趣包命中率, 使得网络中传输效率更高, 达到缓解网络压力、提高用户体验的目的。本文研究的成果有:

(1) 提出了网络拥塞检测机制。在解决网络拥塞的同时, 需要对网络拥塞做出预防, 及时检测网络状态时非常必要的。通过统计中间节点的兴趣包数量, 以此作为指标判断网络拥塞程度, 根据网络拥塞程度将网络状态划分为三种: 网络空闲、网络繁忙、网络拥塞, 并根据这三种网络状态分别对应扩展了 NACK 消息, 向上游链路进行反馈。缓解了 NDN 中网络状态反馈不及时, 程度不准确的问题, 为后续的速率调节做了准备工作。

(2) 提出了基于接收端的速率调节方案。当网络内节点持续不能满足当前需求时, NACK 消息最终会反馈到内容接收端消费者节点, 通过消费者节点降低发送请求的速率, 来解决网络拥塞问题。当检测到网络状态为空闲时, 采用乘性增加的算法, 尽快占据网络空闲带宽; 当检测到网络状态为繁忙时, 采用加性增加的算法, 尽可能避免出现拥塞的现象; 当检测到网络状态为拥塞时, 采用乘性减小的算法, 重新调整拥塞窗口的大小。有效的在接收端对网络状态做出了调整。

(3) 提出了基于路由接口排名的转发策略。为了尽可能让兴趣包通过命中率高的路由接口转发出去, 要构建维护命中率优先级数据结构, 通过命中率优先级表决定到达路由器节点的兴趣包转发的接口。本算法实现了较为客观的接口命中率评估, 以大概率选择命中率高的接口将用户请求转发至可靠的内容源, 在降低网络网络拥塞程度的同时, 周期性的更新 HPT 表, 实时更新不同的内容源路径, 动态调节网络状态。

(4) 在 ndnSIM 模拟平台下, 将本文提出的拥塞控制算法 CCA, 包括拥塞检测机制和基于接收端的速率调节方案和基于路由接口排名的转发策略绑定在

一起，与 NDN 中采用的 AIMD 和最佳路由转发策略进行实验数据方面的对比，从链路带宽利用率、路由节点内容命中率、用户总命中率三个方面进行比较。通过对 Tracer 文件的统计分析，发现在本文的策略在这些性能指标方面有很好的优势，在拥塞控制方面，达到了实时检测网络状态，根据不同的网络状态在接收端调整网络流量，在中间节点提高了命中率，进而提高了网络传输效率，达到了缓解拥塞的目的。

在模拟实验中，通过与 NDN 采用的 AIMD 算法和最佳转发策略进行对比，本文的拥塞控制算法取得了一定的效果，在各个性能指标评价方面都优于 NDN 采用的策略，但是本算法仍存在不足，需要改进，主要有一下方面：

（1）拥塞检测方法。对网络进行拥塞检测时，可以考虑更多的影响因素，例如网络资源的热门程度，资源的访问次数等。可以引入机器学习的方法，构建网络状态预测模型，通过对检测得到的网络状态进行学习，推测下一时刻网络状态的变化趋势。

（2）划分网络状态的方法。本文在拥塞检测中是通过控制发送兴趣包的数量，来模拟网络空闲、网络繁忙、网络拥塞的状态，没有考虑网络突发流量，并不十分准确。

（3）对比实验。本次实验只对用户请求数据量这一个变量进行了控制变量实验，在以后的研究工作中，可以加入多个变量进行实验对比，如开启路由缓存、引入更多路径等，从而验证本文的拥塞控制策略的性能。

参考文献

- [1] Carofiglio, Giovanna, Gallo, Massimo, Muscariello, Luca. Joint hop-by-hop and receiver-driven interest control protocol for content-centric networks[J]. *Acm Sigcomm Computer Communication Review*, 42(4):491.
- [2] Fen Y , Fuchao Y , Xiaobing S , et al. A New Data Randomization Method to Defend Buffer Overflow Attacks[C]// *International Colloquium on Computing, Communication, Control, & Management*. 2010.
- [3] Koponen T, Chawla M, Ermolinskiy A, et al. A data-oriented (and beyond) network architecture [C]// *Proceedings of Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2007)*, Aug 27-31, 2007, Kyoto, Japan. New York, NY, USA: ACM, 2007:181-192.
- [4] 张冰. 电信网网络安全漏洞与补丁管理研究[J]. *电信网技术*, 2006, 000(007):14-18.
- [5] Enhancing NDN feasibility via dedicated routing and caching[J] . Dohyung Kim, Younghoon Kim. *Computer Networks* . 2017
- [6] Zhang, Huajun, An, Xinli, Zhou, Yanyan, 等 . Effect of Oxidative Stress Induced by *Brevibacterium* sp. BS01 on a HAB Causing Species-*Alexandrium tamarense*[J]. *PLoS ONE*, 2013, 8(5):e63018.
- [7] 蔡辉. 内容分发网络技术的研究[D]. 复旦大学, 2013.
- [8] Zhang L , Afanasyev A , Burke J , et al. Named Data Networking[J]. *Computer Communication Review*, 2014, 44(3):66-73.
- [9] 王文静, 雒江涛. 命名数据网络中的一种主动拥塞控制策略[J]. *计算机工程与应用*, 2018, v.54; No.905(10):120-125.
- [10] Mastorakis S , Afanasyev A , Zhang L . On the Evolution of ndnSIM[J]. *Computer Communication Review*, 2017, 47(3):1-1.
- [11] Schneider K, Cheng Y, Zhang B, et al. A practical congestion control scheme for named data networking[C]// 2016.
- [12] Igor Ribeiro, Antonio Rocha, Celio Albuquerque, 等. On the possibility of mitigating content pollution in Content-Centric Networking[C]// *2014 IEEE 39th Conference on Local Computer Networks (LCN)*. IEEE, 2014.
- [13] Gasti, Paolo, Tsodik, Gene, Uzun, Ersin, 等. DoS DDoS in Named-Data Networking[J]. *Acm Sigcomm Computer Communication Review*, 2012, 44(3):66-73.
- [14] Li, Qi, Zhang, Xinwen, Zheng, Qingji, 等. LIVE: Lightweight Integrity Verification and Content Access Control for Named Data Networking[J]. *Information Forensics & Security IEEE Transactions on*, 10(2):308-320.
- [15] Dibenedetto S, Papadopoulos C. Mitigating poisoned content with forwarding strategy[C]//

2016.

- [16] Haiyang Qian, R. Ravindran, Guo-Qiang Wang, 等. Probability-based adaptive forwarding strategy in named data networking[C]// Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on. IEEE, 2013.
- [17] 胡晓艳, 龚俭. 命名数据网络NDN的域间多路径路由机制% Multipath inter-domain routing for named data networking[J]. 通信学报, 036(10):211-223.
- [18] AKM Mahmudul Hoque, Syed Obaid Amin, Adam Alyyan, 等. NLSR: Named-data link state routing protocol[C]// Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking. ACM, 2013.
- [19] A. Afanasyev, P. Mahadevan, I. Moiseenko, 等. Interest flooding attack and countermeasures in Named Data Networking[C]// IFIP Networking Conference, 2013. IEEE, 2013.
- [20] Yi, Cheng, Afanasyev, Alexander, Moiseenko, Ilya, 等. A case for stateful forwarding plane[J]. Computer Communications, 36(7):779-791.
- [21] Carofiglio, Giovanna, Gallo, Massimo, Muscariello, Luca. Joint hop-by-hop and receiver-driven interest control protocol for content-centric networks[J]. Acm Sigcomm Computer Communication Review, 42(4):491.
- [22] Yi, Cheng, Afanasyev, Alexander, Wang, Lan, 等. Adaptive forwarding in named data networking[J]. Acm Sigcomm Computer Communication Review, 42(3):62.
- [23] Sangtae Ha, Injong Rhee, Lisong Xu. CUBIC: A New TCP-Friendly High-Speed TCP Variant[J]. Acm Sigops Operating Systems Review, 42(5):p.64-74.
- [24] 邢思思. 基于RIFVCP与IDWRR的命名数据网络拥塞控制策略研究[D]. 2019.
- [25] 黄胜, 郝言明, 姜良浩, et al. 命名数据网络中基于自适应转发的拥塞控制机制% CONGESTION CONTROL MECHANISM BASED ON ADAPTIVE FORWARDING OVER NAMING DATA NETWORKING[J]. 计算机应用与软件, 2017, 034(012):184-188,294.
- [26] 姜一鸣, 雒江涛. 命名数据网络中的一种冗余控制策略% Redundant control strategy in named data networking[J]. 计算机应用研究, 2018, 035(010):3048-3052.
- [27] Xia Y. One More Bit is Enough[J]. IEEE/ACM Transactions on Networking, 2005, 35(4):37-48.
- [28] A Survey of Information-Centric Networking Research[J]. IEEE Communications Surveys & Tutorials, 2014, 16(2):1024-1049.
- [29] Ahlgren, B, Dannewitz, C, Imbrenda, C, 等. A survey of information-centric networking[J]. IEEE Communications Magazine, 50(7):26---36.
- [30] Daniel Posch, Benjamin Rainer, Hermann Hellwagner. SAF: Stochastic Adaptive Forwarding in Named Data Networking[J]. IEEE/ACM Transactions on Networking, 2015.
- [31] Vasilakos, Athanasios V, Li, Zhe, Simon, Gwendal, 等. Information centric network: Research challenges and opportunities[J]. Journal of Network & Computer Applications, 52(jun.):1-10.
- [32] Abdelberi Chaabane, Emiliano De Cristofaro, Mohamed AN Kaafar, 等. Privacy in Content-Oriented Networking: Threats and Countermeasures[J]. Acm Sigcomm Computer Communication

Review, 43(3):26-33.

- [33]Arianfar S, Nikander P, Eggert L, et al. Contug: A receiver-driven transport protocol for content-centric networks[C]. Proceedings of 18th IEEE International Conference on Network Protocols (ICNP' 10). Kyoto, Japan, 2010.
- [34]Salsano S, Detti A, Cancellieri M, et al. Transport-layer issues in information centric networks[C]. Proceedings of the second edition of the ICN workshop on Information-centric networking. ACM, 2012: 19-24.
- [35]Amadeo M, Molinaro A, Campolo C, et al. Transport layer design for named data wireless networking[C]. Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on. IEEE, 2014: 464-469.
- [36]Braun S, Monti M, Sifalakis M, et al. An empirical study of receiver-based aimd flow-control strategies for ccn[C]. Computer Communications and Networks (ICCCN), 2013 22nd International Conference on. IEEE, 2013: 1-8.
- [37]Carofiglio, Giovanna, Gallo, Massimo, Muscariello, Luca. Optimal multipath congestion control and request forwarding in information-centric networks: Protocol design and experimentation[J]. Computer Networks, 110(dec.9):104-117.
- [38]Danfeng Shan, Wanchun Jiang, Fengyuan Ren. Absorbing micro-burst traffic by enhancing dynamic threshold policy of data center switches[C]// IEEE Infocom -ieee Conference on Computer Communications. IEEE, 2015.
- [39]Feixiong Zhang, Yanyong Zhang, Alex Reznik,等. A transport protocol for content-centric networking with explicit congestion control[C]// 2014 23rd International Conference on Computer Communication and Networks (ICCCN). IEEE, 2014.
- [40]Natalya Rozhnova, Serge Fdida. An effective hop-by-hop Interest shaping mechanism for CCN communications[C]// 2012 Proceedings IEEE INFOCOM Workshops. IEEE, 2012.
- [41]Yu-Ting Yu, R.B. Dilmaghani, S. Calo,等. Interest propagation in named data manets[C]// International Conference on Computing, Networking & Communications. IEEE, 2013.
- [42]Zhang, Feixiong & Zhang, Yanyong & Reznik, Alex & Liu, Hang & Qian, Chen & Xu, Chenren. (2015). Providing explicit congestion control and multi-homing support for content-centric networking transport. Computer Communications. 69. 10.1016/j.comcom.2015.06.019.
- [43]Marica Amadeo, Antonella Molinaro, Claudia Campolo,等. Transport layer design for named data wireless networking[C]// IEEE Infocom -ieee Conference on Computer Communications Workshops. IEEE, 2014.
- [44]Chen, Qingxia, Xie, Renchao, Yu, F. Richard,等. Transport Control Strategies in Named Data Networking: A Survey[J]. IEEE Communications Surveys & Tutorials:1-1.
- [45]Asanga Udugama, Xinyi Zhang, Koojana Kuladinithi,等. An On-demand Multi-Path Interest Forwarding strategy for content retrievals in CCN[C]// Noms IEEE/IFIP Network Operations & Management Symposium. IEEE, 2014.
- [46]Afanasyev, Alexander & Shi, Junxiao & Zhang, Beichuan & Zhang, Lixia & Moiseenko, Ilya

- & Yu, Yingdi & Shang, Wentao & Li, Yanbiao & Mastorakis, Spyridon & Huang, Yi & Abraham, Jerald & Newberry, Eric & Liang, Teng & Schneider, Klaus & DiBenedetto, Steve & Fan, Chengyu & Shannigrahi, Susmit & Papadopoulos, Christos & Pesavento, Davide & Gordon, Nicholas. (2018). NFD Developer's Guide. 10.13140/RG.2.2.28896.38408.
- [47] Yeh, Edmund & Ho, Tracey & Burd, Michael & Cui, Ying & Leong, Derek. (2013). VIP: A Framework for Joint Dynamic Forwarding and Caching in Named Data Networks. ICN 2014 - Proceedings of the 1st International Conference on Information-Centric Networking. 10.1145/2660129.2660151.
- [48] A. Udugama, X. Zhang, K. Kuladinithi and C. Goerg, "An On-demand Multi-Path Interest Forwarding strategy for content retrievals in CCN", 2014 IEEE Network Operations and Management Symposium (NOMS), 2014.
- [49] Andrea Passarella. A survey on content-centric technologies for the current Internet: CDN and P2P solutions[J]. Computer Communications, 35(1):p.1-32.
- [50] Zhangran Chen, Luxi Yang, Yi Li, 等. Cyclobacterium xiamenense sp. nov. isolated from aggregates of Chlorella autotrophica, and emended description of the genus Cyclobacterium[J]. International Journal of Systematic and Evolutionary Microbiology, 2013, 64(Pt 3).
- [51] Kai Lei, Jiawei Wang, Jie Yuan. An entropy-based probabilistic forwarding strategy in Named Data Networking[C]// ICC 2015 - 2015 IEEE International Conference on Communications. IEEE, 2015.
- [52] Carofiglio, Giovanna, Gallo, Massimo, Muscariello, Luca. Optimal multipath congestion control and request forwarding in information-centric networks: Protocol design and experimentation[J]. Computer Networks, 110(dec.9):104-117.
- [53] Ren Y, Li J, Shi S, et al. Congestion control in named data networking - A survey[J]. Computer Communications, 2016, 86(JUL.15):1-11.

攻读硕士学位期间所发表的学术论文

- 1 张丽, 刘晓雨. 一种 NDN 中基于路由接口排名的转发方法. 202010115330.0

致 谢

时光荏苒，白驹过隙，转眼间即将告别校园，走向社会，开始一段新的人生旅程。三年的研究生生活到此就要告一段落了，我的求学生涯也将告一段落。这三年精彩又充实，学到了很多，结识了很多优秀的人，自己也成长了许多。而这都离不开周围老师，学长学姐和朋友们的帮助。

首先要感谢我的研究生导师张丽老师。从刚入学时，张老师的负责与严谨给我留下了深刻的印象，这也时刻鞭策着我在学习上不能掉以轻心。三年来，张老师每周开一次例会，以讲解 PPT 的方式要求我汇报学习和论文进度，在每一次的历练中，老师都给予我很多建议，这极大的提升了我的表达能力和理解能力。另外，在毕设的开题和后续的实验过程以及论文的撰写的过程中，张老师都不遗余力的给予我莫大的帮助，更让我养成了善于思考和关注细节的习惯，这对我以后的学习工作生活都产生了巨大影响。

其次要感谢实验室的同学。感谢谭笑学长在实习和就业方面给了我很多建议和经验作为参考，也很期待以后和谭笑学长在同个工作单位共事；感谢彭兴硕学长在论文实验方面给了我很多学习资料，站在前人的肩膀上让我少走了很多弯路；感谢郑鑫同学在完成毕设过程中和我一起探讨出现的问题，互相帮助共同进步；感谢李晓东、赵韩、潘何益和张强等同学在实验室学习生活中的帮助与建议；感谢舍友特别是徐爽爽同学在三年的研究生生活的陪伴，共同度过了许多欢乐珍贵的时光；感谢同期的很多同学，特别是班干部同学，正是因为有了你们才能让我的研究生生活有条不紊、丰富多彩的进行。

此外还要感谢我的父母，感谢你们对我的培养，在我遇到困难时做我坚强的后盾，无条件支持我，无微不至的照顾我。由于今年疫情的原因，论文撰写的大部分时间是与父母一起度过的，在享受和父母在一起的惬意的同时，也让我更珍惜当下，心怀感恩的走下去。

最后，要感谢老师们对我的论文进行评阅，辛苦您的批评指正，我会根据您的指导意见对论文进行改进与完善！