

基于集中控制的命名数据网络流量调度方法

董谦^{1,2,3}, 李俊¹, 马宇翔^{1,2}

(1. 中国科学院计算机网络信息中心, 北京 100190; 2. 中国科学院大学, 北京 100049;
3. 佛山科学技术学院电子信息工程学院, 广东 佛山 528000)

摘 要: 针对命名数据网络流量全局性优化调度问题, 分析已有工作, 提出一种基于集中控制的方法。所提方法兼顾网络性能与通信开销, 先选择合适节点作为 E-NDN 节点, 再利用控制器根据网内缓存、Interest 包聚合情况和热门内容的流量需求计算相应的多路径转发策略并下发至 E-NDN 节点, 以达到全局性优化的目的。实验结果表明, 所提方法可显著降低最大链路利用率, 提高网络性能, 同时优化代价较小, 控制器与节点间的通信开销略有增加。

关键词: 命名数据网络; 集中控制; 流量调度; 混合网络; 线性规划

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.1000-436x.2018121

Traffic scheduling method based on centralized control in named data networking

DONG Qian^{1,2,3}, LI Jun¹, MA Yuxiang^{1,2}

1. Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China

2. University of Chinese Academy of Sciences, Beijing 100049, China

3. Department of Electronic and Information Engineering, Foshan University, Foshan 528000, China

Abstract: In order to address the global optimization problem for traffic scheduling in named data networking, related works were analyzed, a method based on centralized control was proposed. The proposed method took network performance and communication overhead into account. In the proposed scheme, appropriate nodes would be selected as E-NDN nodes, then the controller calculated the corresponding multi-path forwarding policies and sent them to E-NDN nodes according to the in-network cache, the aggregation of Interest packets, and the traffic demands of popular contents to achieve global optimization. The evaluation results indicate that the proposed method can significantly reduce the maximum link utilization and improve network performance. Simultaneously, the proposed method will not cause a large optimization cost, and communication overhead between the controller and nodes will increase slightly.

Key words: named data networking, centralized control, traffic scheduling, hybrid network, linear programming

1 引言

当今互联网以 TCP/IP 协议族为中心, 但是以 IP 为细腰^[1]的结构已经越来越难以满足互联网的发展趋势, 而以内容为中心的需求使内容分发网络

(CDN, content delivery network) 和对等网络 (P2P, peer-to-peer network) 等技术^[2]得以广泛应用。除此之外, 各种全新设计的互联网架构也引起了研究者的极大兴趣, 其中, 以信息中心网络 (ICN, information centric network) 为代表的未来互联网已

收稿日期: 2017-10-19; 修回日期: 2018-06-05

通信作者: 李俊, lijun@cnic.cn

基金项目: 国家重点研发计划基金资助项目 (No.2017YFB1401500); 国家自然科学基金资助项目 (No.61672490)

Foundation Items: The National Key Research and Development Program of China (No.2017YFB1401500), The National Natural Science Foundation of China (No.61672490)

被视为 5G 的关键技术之一^[3]。ICN 以信息为中心，替换原有的细腰即 IP，它的代表性实例之一是命名数据网络（NDN, named data network）^[4]。

当前，ICN 及 NDN 的研究工作主要集中在体系结构、路由和转发控制、缓存、移动性、安全、应用等方面^[5]。其中，路由与转发控制已有若干重要成果，如自适应转发面^[6]、命名数据链路状态路由（NLSR, named-data link state route）^[7]、双曲路由^[8]等，每个路由器根据转发策略和相关信息决定将 Interest 包转发到哪个端口，自适应转发面维护转发状态。

同时，本文也注意到 NDN 中还有若干有待进一步研究的问题。在此，主要关注 NDN 中应如何调度流量，使全网流量更趋近于合理分布的状态，特别是在有多个消费者和多个生产者的情况下如何进行网络性能优化，如尽可能地最小化最大链路利用率、提高其吞吐量。尽管 NDN 支持多路径转发和负载均衡^[5]，但 Interest 包的转发通常只由消费者或中间节点控制，难以实现全局性优化调度，制约了网络性能的提升。而若要对 NDN 流量做全局性优化调度则必须获取全网信息，因此集中控制的引入十分必要，特别是 NDN 如何与集中控制相结合，如何合理设计集中控制的整体架构，取得性能和开销的平衡。NDN 结合集中控制成为混合网络即网络中存在不同功能的节点时，应考虑如何建立流量模型，选择合适的调度节点并通过这些节点调度流量以及如何快速计算流量分配。软件定义网络（SDN, software defined network）已取得极大成功，其核心思想是集中控制、控制面与数据面分离^[9]，已有一些工作探讨如何将 NDN 与 SDN 相结合^[10-16]。然而，NDN 的名字空间相对 IPv4 来说要大得多，NDN 的转发过程也完全不同于 IPv4，如要借助集中控制对网络流量进行控制，必须充分考虑 NDN 本身的特点，并加以针对性设计。

为此，本文分析了已有的 NDN 多路径转发策略以及 NDN 与 SDN 结合的相关工作，提出了一种基于集中控制的命名数据网络流量调度方法，引入集中控制机制优化 NDN 中的流量转发，结合 NDN 的特点设计整体架构、建立流量模型。本文的主要贡献如下。

1) 分析已有工作，针对 NDN 中流量的全局性优化调度问题，提出一种基于集中控制的方法，设计整体架构和各类设备的功能。

2) 对 NDN 流量调度问题建模和分析，提出一种可行的算法，只在部分节点针对热门内容请求部署合适的转发策略，即可有效对流量进行全局性优化调度，也减轻了控制器与节点间的通信开销。

3) 考虑 NDN 中 Interest 包的作用和特点及网内缓存和 Interest 包聚合等因素，针对已建立的模型提出简便的处理方法。

2 NDN 转发基本流程和相关工作

2.1 NDN 转发基本流程

NDN 中有 2 种不同类型的包：Interest 和 Data。这 2 种包都携带了内容的名字，用以确认和区分不同的内容。消费者向网络发送 Interest 包来请求相应的内容，一旦 Interest 包到达拥有该内容的节点，节点会沿反向路径将包含名字和内容的 Data 包返回给消费者，同时还携带有内容生产者的签名^[5]。

每个 NDN 路由器都会维护 3 种数据结构：待定 Interest 表（PIT, pending Interest table）、转发信息库（FIB, forwarding information base）和内容存储（CS, content store）。同时，路由器还要维护转发策略模块，以决定 Interest 包如何转发。PIT 存储了路由器已转发但还未返回相应 Data 包的所有 Interest 列表，PIT 中的每个条目记录相应的数据名字和出入端口；而 CS 存储了本地缓存的所有内容，如果收到相应的 Interest 请求，则沿原端口返回相应内容，如果 CS 中未找到，则路由器检查 PIT，如有相关记录则添加其端口，如无则继续查找 FIB，并按相应策略转发，FIB 中一个名字可对应多个接口；如果收到多个完全相同的 Interest 包，则只转发第一个 Interest 包，Interest 包的聚合机制是 NDN 中防止环路及广播风暴特性的基础^[5]。

反过来，对于 Data 包，情况就简单多了。当 Data 包抵达时，NDN 路由器会查找 PIT，按照其条目中的相应端口进行转发，并将内容存入 CS，移除 PIT 相应条目，此时，相应 Interest 请求已经得到满足，Data 包则沿着 Interest 的反向路径转发^[5]。

显然，在 NDN 转发模型中，Interest 包由消费者驱动，由于 Data 包的转发严格按照 Interest 包的反向路径，故 NDN 可通过控制 Interest 包的转发路径来管理 Data 包的转发路径，从流量调度的角度来看，Interest 包不直接传递内容，可将其看作一种信令，它为流量调度和路径控制提供了简便的方式。

2.2 相关工作

NDN 在路由和转发方面与传统网络非常不同,为此,研究者们进行了相关研究。Yi 等^[6]提出自适应转发机制,通过观察 Interest 和 Data 包的双向流量来检测网络问题、多路径和环路;Hoque 等^[7]提出了 NLSR,使用 Interest 和 Data 包来传播路由更新,可对每个名字前缀生成转发选项的排序,以服务于自适应转发策略;Lehman 等^[8]尝试将双曲路由应用于 NDN 中,以解决路由可扩展性问题,虽然取得了和 NLSR 较为接近的效果,却极大降低了路由协议的开销;Posch 等^[17]提出了随机自适应转发来模仿水管系统,能够在无路由信息更新的情况下识别路径,避开链路故障和网络瓶颈。以上工作使 NDN 具备良好的多路径转发特性。

基于此,一系列工作研究了 NDN 中的多路径转发策略,主要是在多路径转发的情况下,如何使流量合理分配到不同的路径上,从而获取更好的网络性能。Carofiglio 等^[18]考虑最大限度提高用户吞吐量和最小化整体目标网络成本,从双重全局优化问题出发,提出分布式拥塞控制策略动态调整包的转发;Detti 等^[19]建模分析了 ICN 中的多路径转发策略,并介绍了与之相关的 5 种方案;Xin 等^[20]基于本地内容的流行度统计,对包进行调度以提高缓存效率和避免拥塞,提出缓存替换算法;Udugama 等^[21]针对服务质量(QoS, quality of service)需求,提出一种基于权重的轮转分配机制,只考虑不相交路径,发现路径并分配流量;Kerrouche 等^[22]同样针对服务质量需求,动态获取 QoS 参数并据此调整转发策略。表 1 总结了上述工作。

总之,上述工作通过不同方式获得多条路径上的相关参数,并依此调度流量请求;上述工作还表明,多路径转发对于网络性能的提升有重要作用,特别是将流量请求按照合适的方式分配到不同路径,能提高网络吞吐量、减少网络拥塞。

然而,上述工作都是在消费者或中间节点处对

具体内容请求的多路径转发进行控制,均未引入集中控制,因此难以针对多个消费者和多个生产者同时存在需要对流量进行全局性优化调度的情况。虽然缓存是 NDN 的最重要特征之一,但部分工作在建模时并未考虑此因素。

与此同时,一系列工作研究了集中控制机制和 NDN 的结合。

1) 从功能的角度,Torres 等^[10]首先提出基于控制器的 NDN 路由方案,控制器可获取拓扑及计算路由,存储命名数据的位置;Chanda 等^[11]设计了软件定义信息中心网络的框架,并简单讨论了在此框架上部署流量工程任务;Bacher 等^[12]使用控制器为未知内容 Interest 包计算路径,并在链路拥塞的情况下查找替代路径,根据用户需求提前部署相应热门内容以减轻核心网络负担;Gao 等^[13]针对软件定义 ICN 提出一种用于域内通信的可扩展区域分层架构以解决控制平面可扩展性问题,支持对网络资源和内容资源的感知,保证有效的兴趣匹配和资源适配。

2) 从部署的角度,Salsano 等^[14]在实验床进行测试,验证了基于 SDN 的 ICN 的可行性;Van 等^[15]提出的 NDNFlow,实际上是以 NDN 客户端插件的形式将 NDN 功能部署在原有的 SDN 设备上;Mahmood 等^[16]为缓解控制器负担重的问题,使用有状态 S/N-DN 设备,在 NDN 中部署集中控制。

上述工作从不同角度讨论了集中控制与 NDN 的结合,此时控制器可采集和分发各类信息、下发动作等;若要实际部署,相关开销是必须考虑的问题。

然而,上述工作并未讨论如何基于集中控制方法在多路径转发的情况下对流量进行全局性优化调度;NDN 相比 IPv4 大得多的命名空间也会显著增加控制器的负担,如果在所有节点部署 SDN 和 NDN 功能、控制所有流量,则网络的可扩展性难以提高。

表 1 几种多路径转发策略比较

方案	优化目标	主要机制	路径产生方式	有无集中控制	建模有无考虑缓存
文献[18]的方案	网络性能	根据待定 Interest 数量和端口负载调节请求转发	路由/预配置	无	有
文献[19]的方案	网络性能	测量路径往返时间,调整请求转发的端口权重	路由/预配置	无	无
文献[20]的方案	网络性能、缓存性能	根据内容流行度统计,调度请求和替换缓存	路由/预配置	无	有
文献[21]的方案	网络性能	测量路径往返时间,调整请求转发的端口权重	发现	无	无
文献[22]的方案	网络性能	更新路径 QoS 参数,动态调整请求转发端口	路由/预配置	无	无

3 整体架构

在通常的NDN中, Interest包的转发是由消费者或中间节点根据名字、转发表和转发策略决定的, 过程上和传统网络中基于链路状态的路由协议相似。如相关工作所述, 考虑多个消费者和多个生产者同时存在的情况, 即考虑对流量进行全局性优化调度的需求, 则有必要引入集中控制; NDN不同于IPv4, 其与集中控制结合需考虑NDN本身的特点, 还需考虑相关开销。因此, 本文从多路径转发和集中控制的角度出发, 结合NDN流量调度通过Interest包来实现的做法, 在部分节点针对热门内容请求部署合适的转发策略, 可有效对流量进行全局性优化调度, 这也减轻了控制器与节点间的通信开销。本文提出一种基于集中控制的命名数据网络流量调度方法, 控制器仅控制部分节点对部分内容请求的转发, 其他内容仍然采用默认的转发机制, 面向多个消费者和多个生产者, 对网络性能如最大链路利用率进行全局性优化。

基于对中国科技网等实际网络架构、拓扑和业务部署情况的分析和抽象, 对于汇聚各子网和终端用户, 本文将其虚拟为若干个消费者, 每个虚拟消费者对内容的请求可认为是相对应的所有终端用户的内容请求经过Interest包聚合过程后发送给其汇聚节点的内容请求汇总, 因此每个虚拟消费者可用单条链路接入相应的汇聚节点; 对于服务器等内容提供设备亦然, 可虚拟为若干个生产者, 如所有域外的服务器可通过相应链路接入域间互连节点, 域内的服务器也可通过相应链路接入域内节点。上述方式处理实际拓扑以后, 流量调度问题即可方便地表示为有向图中的多商品流(MCF, multi commodity flow)问题^[23]。

根据文献[24]统计的网站访问热度以及内容请求服从类Zipf分布可知, 多数流量产生自用户对热门内容的请求, 如视频网站、门户网站、搜索引擎等, 优化这部分前缀转发的效果较为明显, 其他内容仍然依照默认的转发流程, 这样不仅可降低采集相关信息的开销, 也可大幅减少控制器下发规则或转发策略的数量。

整体架构示意如图1所示, 主要由控制器、2类中间设备(普通NDN节点和增强NDN节点即E-NDN节点)和2类端设备(虚拟消费者和生产者)构成。

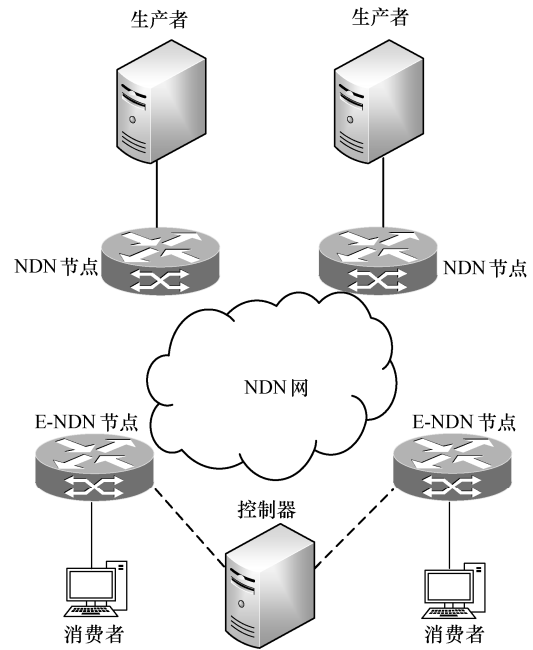


图1 整体架构示意

各类设备的主要功能介绍如下。

1) 控制器需与网内节点建立连接, 其作用是获取全网拓扑, 以一定时间间隔通过NETCONF^[25]等南向接口协议周期性采集网内节点的信息, 如FIB、流量情况、缓存命中率、待定Interest包数量等, 并及时处理各E-NDN节点发送的流量调度需求, 进行优化计算并将结果转换为转发规则或转发策略下发给相应E-NDN节点。

2) 普通NDN节点支持网内缓存, 遵循NDN的转发流程, 依照转发表和转发策略对Interest包进行转发。一般情况下, 普通NDN节点的转发表和转发策略都比较固定, 这也使网络拓扑未发生变化、链路未发生拥塞或故障时, Interest包的转发路径及与之对应的Data包转发路径是相对稳定的。控制器对普通NDN节点的操作可认为是只读的, 而普通NDN节点需要根据控制器的采集需求, 及时采集并发送相应信息。

3) E-NDN节点除了有普通NDN节点的功能以外, 还应根据待定Interest包的数量, 将较为热门内容的流量调度需求发送给控制器, 控制器处理完毕后, 接收控制器下发的转发规则或转发策略; 有多个下一跳时, 能将待优化转发的那部分Interest包按照控制器决定的方式分别转发到多条路径; 转发后的处理流程和普通NDN节点相同, 以保证返回的Data包沿着相应Interest包的反向路径转发。

4) 消费者主要产生对内容的请求即Interest包; 生产者满足这些Interest包, 返回相应内容的Data包。

4 模型建立与问题分析

4.1 模型建立

通常网络流量优化调度的目标是降低最大链路利用率。以此优化目标为例, 本文首先建立一般化的流量模型, 该模型也可推广用于同时存在其他优化目标和约束条件的情况, 如全局最小费用等。

首先, 在流量模型中需定义有向图 $G=(N, E)$, N 为所有节点的集合, E 为所有边的集合, 每条边即 2 个节点间直连链路的权重集合为 W , 容量集合为 C , 对于节点 i 到 j 的直连链路可记为 w_{ij} 和 c_{ij} , 若节点 i 到 j 的直连链路为 e , 亦可记为 w_e 和 c_e 。节点 i 的直连节点即下一跳集合为 NH_i , 显然, 节点 i 的上一跳集合也为 NH_i , 从节点 i 到节点 j 的直连链路 e 上的流量记为 y_e , 衡量单位时间内从节点 i 传输到节点 j 的数据量, 显然有

$$0 \leq y_e \leq c_e, \forall e \in E \quad (1)$$

式(1)是链路容量条件。针对多个消费者和多个生产者同时存在的情况即多商品流问题, 将源节点和目的节点记为 s, t , 设所有流的源节点 s 集合为 S , 每个 s 对应的所有目的节点 t 集合为 T_s , 显然, S, T_s 都是 N 的子集; 节点 s, t 间的流量需求记为 d_{st} , 所有流量需求的集合以 D 表示。为方便表达, 记任意节点 i, j 间的流量需求为 d_{ij} , i 对应的目的节点集合表示为 T_i , 若 i 不属于 S , 则 T_i 为空集, 有

$$d_{st} \geq 0, \forall s \in S, t \in T_s \quad (2)$$

$$d_{ij} = 0, \forall i \in N, j \in N - T_i \quad (3)$$

式(2)和式(3)表明, 除了节点 s, t 间以外, 其余节点间没有流量需求。因此, 不考虑环路时(形成环路的流量是无效流量, 无助于满足流量需求)满足所有流量需求的网络总流量 $Traffic$ 为

$$Traffic = \sum_{s \in S} \sum_{j \in T_s} d_{sj} \quad (4)$$

为了满足流量需求, 建立完整的流量模型, 还需要写出相应的流量平衡约束条件, 然而处理 NDN 流量优化调度问题时, 由于 NDN 网内缓存和 Interest 包聚合的特点, 流量平衡约束条件较为复杂。考虑到 d_{st} 是从 s 到 t 的流量需求, 流量平衡约束条件实际上体现在起点为 s 终点为 t 的路径上, 且根据第 3 节, 网络中只有部分节点可调度, 因此, 可预先计算好 s, t 间的可选路径, 然后用合适方式

表达以便求解如何将流量需求合理分担在各条可选路径上, 还可预先排除存在环路的路径, 这是因为存在环路的路径必然不是最优解。

设 m 为集合 E 的元素个数即所有边的数量, 对每条边编号, 记为 e_l , 编号 $l=1, 2, \dots, m$, 此时 s, t 间的任意一条路径都记为由 m 个元素组成的一维数组 p , 可选路径集合为 P_{st} , 假如从 s 到 t 且经过此路径的流量为 1, 则 p 的元素值为此时各边上的流量大小, 显然在路径没有重复使用某条边的情况下, 当路径经过边 e_l 时, 数组中的第 l 个元素值为 1, 而不经过的边对应的数组中的元素值为 0, 流量平衡约束条件即反映在此数组中, 即使考虑 NDN 中网内缓存和 Interest 包聚合, 也只需采取 4.2.2 节中的做法。值得注意的是, 若考虑等价多路径负载均衡的情况, 且均衡点只是普通 NDN 节点, 依照固定的均衡策略转发, 只需根据负载均衡后的流量分担情况设置相应数组中的元素值, 例如, s, t 间有 2 条等价路径, 则 2 条等价路径各负担一半流量, 对于不重叠的部分, 元素值为 $\frac{1}{2}$, 对于重叠的部分则仍为 1,

可将此时的流量平衡情况记为一个一维数组, 假设一条路径经过 e_1, e_2 , 另一条等价路径经过 e_1, e_3 , 此时数组的第一个元素值为 1, 第二、三个元素值为 $\frac{1}{2}$, 不需要记为 2 个数组, 原因在于这 2 条路径的流量分担比例是固定的。不过, 若均衡点是 E-NDN 节点, 其具备灵活的流量调度能力, 则相关各条等价路径的流量分担比例并不固定, 因此需将相关各条等价路径记为不同的数组参与计算。

为简化讨论, 本文后续不考虑普通 NDN 节点支持等价多路径负载均衡时的情况, 如果普通 NDN 节点作为均衡点有多条等价路径, 计算时只取其中一条, 也排除路径重复使用某条边的情况。设从 s 到 t 的路径 p 分担的流量需求为 x_p , 边 e 编号为 l , p 中对应的元素值为 $p[l]$, 显然 $p[l]=0$ 或 1, 且有

$$x_p \geq 0, \forall s \in S, t \in T_s, p \in P_{st} \quad (5)$$

$$d_{st} = \sum_{p \in P_{st}} x_p, \forall s \in S, t \in T_s \quad (6)$$

$$y_e = \sum_{s \in S} \sum_{t \in T_s} \sum_{p \in P_{st}} x_p p[l], \forall e \in E, l \text{ 为 } e \text{ 的编号} \quad (7)$$

式(5)是由 x_p 的定义决定的, 式(6)表明 s, t 间的流量需求得到满足, 式(7)是满足所有流量需求时链路 e 上的流量。

再定义链路利用率为 λ_e ，最大链路利用率为 θ ，显然 $0 \leq \theta \leq 1$ ，有

$$\lambda_e = \frac{y_e}{c_e} \quad (8)$$

$$\theta = \max_{e \in E} \{\lambda_e\} \quad (9)$$

此时，式(1)变为

$$0 \leq y_e \leq \theta c_e, \forall e \in E \quad (10)$$

通常，对于多商品流问题的流量模型而言，主要优化目标是最大化 *Traffic* 和最小化 θ ，即

$$\max \text{Traffic} \quad (11)$$

$$\min \theta \quad (12)$$

对于式(11)，约束条件是式(1)~式(7)；若 D 已知，则 *Traffic* 亦确定，此时，对于式(12)，约束条件是式(5)~式(7)和式(10)，且 $0 \leq \theta \leq 1$ 。

式(12)是 minimax 问题^[26]，当路径数量较多时求解较为困难。为此，可引入临时变量 θ' ，使用合适方法（如二分法，具体做法参见 4.2.3 节算法 3）控制 θ' 值并试探求解，根据其精度要求设定试探求解的次数为 K ，最后求得的解 x_p 和相应 θ 即为近似最优解。按此方法在每次试探求解时， θ 的值实际上已经确定，体现在约束条件中，本文还可根据需要添加其他优化目标和约束条件，在必要时还可设置不同的权重进行多目标优化计算，以求解出同时满足多个优化目标的近似最优解。

4.2 问题分析

模型建立过程说明处理此优化问题前需得出可用路径集合，一旦 E-NDN 节点确定，则相应的路径集合也可确定，因此需选择合适的节点作为 E-NDN 节点。NDN 具有 Data 包与 Interest 包一一对应且 Data 包严格按照 Interest 包反向路径转发的特点，因此本文只讨论对 Interest 包流量的优化调度，对 Interest 包转发优化等效于对 Data 包转发优化。此时，节点 s 可视为虚拟消费者单条链路接入的汇聚节点，节点 t 为虚拟生产者接入的节点，假设每个虚拟消费者即相应的节点 s 都需要一个 E-NDN 节点，用于调度此虚拟消费者的内容请求；NDN 最重要的特点是网内缓存和 Interest 聚合，因此，需采取合适的做法将这 2 个因素反映到流量模型中；网络流量分为不可调度流量和可调度流量，需以前者为背景流量，考虑后者的调度问题。另外，本文还假设 Data 包中的内容实际上是请求内容的

数据块即 Chunk，每个 Chunk 大小相同。

4.2.1 E-NDN 节点和可选路径选择

本文首先选取合适的候选路径数量 k ，并用 k 最短路径算法^[27]对每个节点 s 均求出其到 T_s 中所有节点 t 的前 k 条最短路径，输入为 G 、 S 、 T_s 、 W ，输出为路径集合 $P_{st\text{-init}}$ ，显然， P_{st} 是其子集；然后用最短路径算法求得每一组 s 、 t 的最短路径，并对每个节点 s 均求出其到 T_s 中所有节点 t 最短路径中重合的部分，将这些重合的节点标记下来，显然，重合部分的节点至少包含一个节点即节点 s 本身，调度节点 s 内容请求的 E-NDN 节点将在这些节点中选择，原因在于节点 s 到 T_s 中所有节点 t 的流量默认情况下均要经过这些节点，如果 E-NDN 节点不在这些节点上选择则难以有效调度流量，在 E-NDN 节点之外，Interest 包转发仍然按照最短路径方式。

考虑节点 s 到 T_s 中所有节点 t 的 $P_{st\text{-init}}$ ，结合候选节点 i 的 NH_i ，可针对 NH_i 中的每个节点在 $P_{st\text{-init}}$ 中得到一条经过此节点的 s 、 t 间的最短路径，还应排除有环路的路径；对于所有候选节点 i ，都按上述原则筛选 $P_{st\text{-init}}$ ，得到可选路径 P_{st-i} ，然后由 P_{st-i} 计算此时节点 s 到 T_s 中所有节点 t 的总流量最大值 $Traffic_{s\text{-max}}(i)$ ，对所有候选节点取计算结果最大者作为 s 对应的 E-NDN 节点，若有多个则在其中取包括 s 在内的离 s 跳数最少者，相应的 P_{st-i} 即为后续计算使用的 P_{st} ， $Traffic_{s\text{-max}}(i)$ 最大值即为 $Traffic_{s\text{-max}}$ 。

上述过程的算法如下所示。

算法 1 E-NDN 节点和 P_{st} 选择

- 1) 输入 G 、 S 、 T_s 、 W 、 C
- 2) for each $s \in S$ do
- 3) for each $t \in T_s$ do
- 4) k 最短路径计算得 $P_{st\text{-init}}$;
- 5) 最短路径计算;
- 6) end for
- 7) 求 s 到 T_s 中所有 t 最短路径的重合节点;
- 8) $Traffic_{s\text{-max}} \leftarrow 0$;
- 9) for each $i \in$ 重合节点 do
- 10) 取 NH_i 并根据 NH_i 筛选 $P_{st\text{-init}}$ 得 P_{st-i} ;
- 11) 根据 P_{st-i} 求 $Traffic_{s\text{-max}}(i)$;
- 12) if $Traffic_{s\text{-max}}(i) > Traffic_{s\text{-max}}$ then
- 13) s 对应的 E-NDN 节点 $\leftarrow i$;
- 14) $P_{st} \leftarrow P_{st-i}$;
- 15) $Traffic_{s\text{-max}} \leftarrow Traffic_{s\text{-max}}(i)$;
- 16) else if ($Traffic_{s\text{-max}}(i) = Traffic_{s\text{-max}}$) \wedge

(i 比 s 对应的 E-NDN 节点离 s 跳数更少) then

- 17) s 对应的 E-NDN 节点 $\leftarrow i$;
- 18) $P_{st} \leftarrow P_{st-i}$;
- 19) else
- 20) end if
- 21) end for
- 22) end for

控制器采集网络相关信息后即可执行此算法, 计算完成后如网络相关信息不发生变化则不需要重新计算, 如发生变化则应重新执行。由于缓存情况相对其他信息来说变化更加频繁, 为使计算结果稳定, 上述过程一般不考虑缓存因素, 此算法中也未体现; 如需考虑缓存因素, 则每次计算 $Traffic_{s-\max}(i)$ 前按照 4.2.2 节中的做法处理即可。

4.2.2 网内缓存与 Interest 聚合

NDN 不同于传统网络的特点是网内缓存和 Interest 聚合。网内缓存有多种部署方式, 其中一个重要指标是缓存命中率, 通常缓存命中率分别在各个节点上统计, 设节点 i 上的缓存命中率为 h_i , 对于只是流经节点 i 的 Interest 包, 流出 i 的流量是流入流量的 $1-h_i$ 倍^[28]; 对于在此产生的 Interest 包, 考虑到本文简化了场景, 实际上节点 s 是虚拟消费者单条链路接入的汇聚节点, 因此也应考虑缓存的影响, 由源节点产生的流量也应乘以 $1-h_s$ 。

流量平衡约束条件体现在路径数组 p 中, 可根据缓存情况修正 p 中相应元素的值以更好地反映实际情况, 为了不改变 P_s , 对每一个 p 可设置一个缓存系数数组 q_p , 其也是由 m 个元素组成的一维数组, 与 p 一样对应 E 中的所有边, 其计算过程如下: 首先, 使 $q_p = p$, 由于 p 的非零元素值为 1, 对于 q_p , 此时只需将以 s 为起点的路径上的边对应的数组元素值设为 $1-h_s$, 将以第二个节点 i 为起点的路径上的边对应的数组元素值设为 $(1-h_s)(1-h_i)$, 依次类推, 直到将以倒数第二个节点 j 为起点的路径上的边对应的数组元素值设为 $(1-h_s)(1-h_i)\cdots(1-h_j)$ 为止。优化计算可使用 q_p 代替 p , 如果缓存命中率发生变化则更新 q_p 的值。值得一提的是, 对于等价多路径负载均衡的情况, 计算前将其数组还原为相应的若干个等价且非零元素值为 1 的路径数组, 然后按上述方法分别计算各自的缓存系数数组, 再分别乘以相应路径的流量分担比例后相加, 其结果即为这个数组对应的缓存系数数组。

上述过程的算法如下所示。

算法 2 缓存系数数组 q_p 计算 (以 p 为例)

- 1) 输入 p 、 h_i
- 2) $b \leftarrow 1$;
- 3) $q_p \leftarrow p$;
- 4) $i \leftarrow p$ 的起点;
- 5) for $temp = 1$ to p 中非零元素的数量 do
- 6) 查找 p 中起点为 i 的路径上的边, 其编号为 l , 对应 q_p 第 l 项, 相应元素值表示为 $q_p[l]$;
- 7) $q_p[l] \leftarrow b(1-h_i)$;
- 8) $b \leftarrow q_p[l]$;
- 9) $i \leftarrow$ 这条边的终点;
- 10) end for

控制器采集各节点缓存命中率信息后, 即可对路径数组执行此算法, 如果各节点的缓存命中率发生变化, 则应重新执行。

同样地, Interest 包聚合也会影响流量平衡。当不同消费者的请求为独立事件时, 通常难以对 Interest 包的聚合情况进行定量分析, 只能确定聚合后的流量不大于聚合前的流量, 可将其全部视为不聚合流量; 当不同消费者同时请求同样内容且不调度时, Interest 包的聚合情况是已知的, 可对 S 中所有节点 s 的 T_s 求并集 T , 以 T 中所有节点 t 为根, 分别求其到对应的所有 s 的最短路径树 r , r 也可由 m 个元素组成的一维数组表示, 由于聚合原因, r 中每个非零元素的值也为 1, 所有 r 的集合为 R 。如有必要考虑缓存因素, r 也可设置缓存系数数组 q_r 并按类似方法处理, 对于树上除了根以外的节点 i , 当计算以 i 为起点的路径树上的边对应的数组元素值时, 若 i 是根 t 对应的 s 则设为 $1-h_i$, 若不是则先比较以 i 为终点的所有路径树上的边对应的数组元素值之和与 1 的大小, 再取较小者乘以 $1-h_i$ 。

4.2.3 不可调度流量与可调度流量

不可调度流量在所有 NDN 节点均采用默认的转发路径, 而可调度流量必须在 E-NDN 节点才能通过配置相应转发策略来改变其默认转发路径; Interest 聚合要求不同消费者同时请求同样内容, 这种情况在现实中很少出现且难以定量分析, 因此除了视频会议等 Interest 聚合的典型应用以外, 其他请求均可视为不聚合流量; 本文只考虑调度不聚合流量, 可聚合流量均不可调度。据此区分不聚合不可调度流量、可聚合不可调度流量、不聚合可调度流量如下: 除 Interest 聚合应用外, 不属于热门内容的内容请求可视为不聚合不可调度流量; Interest 聚合

应用的内容请求可视为可聚合不可调度流量；除 Interest 聚合应用外，热门内容可视为不聚合可调度流量。针对这 3 种情况， s 、 t 间的流量需求 d_{st} 可分为不聚合不可调度流量需求 d_{st-I} 、可聚合不可调度流量需求 d_{st-II} 与不聚合可调度流量需求 d_{st-III} 。再将所有 s 、 t 间的最短路径表示为 u ，所有 u 的集合为 U ，路径 u 分担来自相对应消费者到生产者的不聚合不可调度流量需求为 x_u ，相应的缓存系数数组为 q_u ；路径树 r 分担的来自相对应消费者到生产者的可聚合不可调度流量需求为 x_r ；路径 p 分担的来自相对应消费者到生产者的不聚合可调度流量需求仍表示为 x_p ；显然 d_{st-I} 即为相对应的 x_u ， d_{st-II} 即为相对应的 x_r ，式(6)中的 d_{st} 改写为 d_{st-III} ，而且有

$$x_u \geq 0, \forall u \in U \quad (13)$$

$$x_r \geq 0, \forall r \in R \quad (14)$$

$$d_{st} = d_{st-I} + d_{st-II} + d_{st-III}, \forall s \in S, t \in T_s \quad (15)$$

计算时若考虑缓存因素，如前所述边 e 的编号记为 l ，则缓存系数数组中边 e 对应的元素值为 $q_p[l]$ 、 $q_u[l]$ 、 $q_r[l]$ ，则式(7)中 y_e 改写为

$$y_e = \sum_{s \in S} \sum_{t \in T_s} \sum_{p \in P_{st}} (x_p q_p[l]) + \sum_{u \in U} (x_u q_u[l]) + \sum_{r \in R} (x_r q_r[l]) \quad (16)$$

最终，设定优化目标并计算完成后输出的结果为 x_p 。可给出主要优化目标为 $\min \theta$ 时求解的算法如下所示。

算法 3 主要优化目标为 $\min \theta$ 时求解 x_p

- 1) 输入 d_{st-I} 、 d_{st-II} 、 d_{st-III} 、 q_p 、 q_u 、 q_r 、 G 、 S 、 T_s 、 C 、 P_{st} 、 U 、 R
- 2) $\theta' \leftarrow 1$;
- 3) $\theta'' \leftarrow 1$;
- 4) $\theta''' \leftarrow 0$;
- 5) for $temp = 1$ to 试探求解次数 do
- 6) 可设定其他优化目标，以 θ' 试探求解 x_p ;
- 7) if x_p 有解 then
- 8) $\theta \leftarrow \theta'$;
- 9) $\theta'' \leftarrow \theta'$;
- 10) $\theta' \leftarrow \theta''' + \frac{\theta'' - \theta'''}{2}$;
- 11) else
- 12) $\theta''' \leftarrow \theta'$;
- 13) $\theta' \leftarrow \theta''' + \frac{\theta'' - \theta'''}{2}$;

14) end if

15) end for

其中， P_{st} 使用算法 1、 U 使用最短路径算法、 R 使用最短路径树算法得出， q_p 、 q_u 、 q_r 按照 4.2.2 节中的做法得出， d_{st-I} 、 d_{st-II} 、 d_{st-III} 由控制器处理完 E-NDN 节点发送的调度需求及控制器采集的相关信息后给出，最后 x_p 变为转发策略下发相关节点。

4.2.4 优化代价评估

本文还通过相应路径长度拉伸的程度 (path stretch) 评估优化代价，通常定义为路径的实际长度与最短路径长度的比值^[29]。那么可设路径 p 和 u 对应的路径长度为 z_p 和 z_u ， s 、 t 间的最短路径长度为 z_{st} ，由于可聚合流量不可调度，本文只考虑不聚合流量，包括不聚合可调度流量和不聚合不可调度流量，全网的不聚合流量路径长度拉伸的程度用 Z 表示，若不考虑缓存因素，则有理论值 Z 为

$$Z = \frac{\sum_{s \in S} \sum_{t \in T_s} \sum_{p \in P_{st}} (z_p x_p) + \sum_{u \in U} (z_u x_u)}{\sum_{s \in S} \sum_{t \in T_s} (z_{st} \sum_{p \in P_{st}} x_p) + \sum_{u \in U} (z_u x_u)} \quad (17)$$

若要考虑缓存因素，则只能实际统计调度与未调度情况下的平均路径长度，其比值即为实测值。

4.2.5 控制器与节点间的通信开销

与 SDN 类似，控制器与节点间的通信开销主要有以下 2 类。

1) 周期性信息采集。此时，通信开销取决于信息采集周期、信息采集量、节点数量，若周期、采集量、节点数都固定，则此类开销也固定。

2) 流量调度需求发送和转发策略下发，用于计算的信息采集 (如需要)。此时，通信开销取决于流量调度需求的个数、每个流量调度需求对应的转发策略、每个流量调度需求对应的信息采集量 (若需要)。流量调度需求表现为相应前缀的流量调度需求，此类开销可认为正比于流量调度需求的个数，也即正比于各 E-NDN 节点上需调度的前缀数量。假设内容前缀总数为 A 个，由于内容请求服从类 Zipf^[24] 分布，前缀已在各 E-NDN 节点按照相应内容的热门程度即待定 Interest 数量从大到小排序，表示为 $Prefix_v$ ，其中，序号 $v = 1, 2, 3, \dots, A$ ，分布系数为 α ，则每个 $Prefix_v$ 被请求的概率为

$$CP_{Prefix}(v) = \frac{\left(\sum_{v=1}^A \frac{1}{v^\alpha} \right)^{-1}}{v^\alpha} \quad (18)$$

若 A 和 α 确定, 则 $CP_{Prefix}(v)$ 也确定。当需调度内容的流量需求占全部流量需求的比例为 β 时, 显然有 $0 \leq \beta \leq 1$; 记需调度 $Prefix_v$ 的序号 v 集合为 V , 则有

$$\beta = \sum_{v \in V} CP_{Prefix}(v) \quad (19)$$

此类开销正比于 V 中元素 v 的数量, 根据式 (18), 当 v 越小, $CP_{Prefix}(v)$ 越大, 因此如要减小此类开销, 各 E-NDN 节点应优先请求调度流量需求较大的热门内容。

5 仿真实验

本文使用基于 NS-3^[30] 的开源工具 ndnSIM^[31] 进行仿真实验, 版本是 2.3。ndnSIM 能部署消费者及生产者应用, 配置相关前缀和路由, 支持多种常见的缓存替换策略; 自 2.0 版本起, ndnSIM 的转发模块更新为已用于实际硬件开发的 NFD (NDN forwarding daemon), 这不仅使其能更好地支持最短路径、多播、NCC、客户端控制等转发策略, 还使仿真结果更接近真实情况。

实验拓扑取自 GÉANT, 共 23 个汇聚层节点, 它们之间的直连链路共 37 条, 考虑双向则共有 74 条边; 设定消费者 3 个, 生产者 2 个, 均为 GÉANT 中的节点, 节点间的流量需求信息取自此拓扑的公开数据集^[32], 共取 4 个时间段的流量情况分别作为 4 个场景进行仿真。根据第 4 节的讨论, 仿真首先需设定一些前提: 每个节点上部署缓存及相应的缓存替换策略; 用户的一般内容请求即不聚合流量服从类 Zipf 分布, 而聚合流量需采用恒定速率, 这是为了满足 Interest 包聚合的条件; 链路的权重均为 1, 带宽为 100 Mbit/s, 时延为 10 ms; Data 包的大小为 1 024 B; 每次实验进行 150 s, 前 50 s 为预热时间, 采用后 100 s 的数据。另外, 参考相关文献[28, 33], 基本参数配置如表 2 所示。

参数	默认值	变化范围
内容总数	80 000	无
缓存容量/时隙	20	无
缓存替换策略	最不经常使用 (LFU)	无
Zipf 分布参数 α	0.9	0.8~1.1

4 种场景中, 消费者分别向生产者发起的内容请求速率如表 3 所示。

本文分别评估缓存的影响、与其他方法对比、不同场景下的效果和优化代价、有流量聚合时不同场景下的效果、控制器与节点间的通信开销。

1) 缓存的影响

当全部流量为不聚合流量时, 针对场景 1, 改变 Zipf 分布的 α 值, 分别取 0.8、0.9、1.0、1.1, α 值越大代表请求的内容越集中, 特别是在 LFU 缓存替换策略下, 缓存命中率相对其他策略在其他参数相同的情况下更高; 此外, 当 α 值为 1.0 和 1.1 时, 还在计算时分别考虑缓存和不考虑缓存; 取不聚合流量中可调度流量比例为 0.2、0.3、0.4、0.5、0.6、0.7、0.8, 不可调度流量仍按照最短路径转发, 按上述条件计算结果并仿真, 统计其最大链路利用率, 分别与相同参数下全部流量为不可调度流量即不聚合流量中可调度流量比例为 0 时的默认情况下最大链路利用率取比值, 最后用不考虑缓存计算得出的理论值作为参照, 结果如图 2 所示。

由图 2 可知, 采用本文方法后最大链路利用率相对默认情况下的比值有不同程度的降低, 可分为 2 个阶段: 阶段 1, 在不聚合流量中可调度流量比例达到 0.5 以前, 其比例越高则最大链路利用率相对默认情况下的比值降低幅度越大; 阶段 2, 在不聚合流量中可调度流量比例达到 0.5 以后, 最大链路利用率相对默认情况下的比值接近最优, 即便再提高不聚合流量中的可调度流量比例, 最大链路利

表 3 各场景内容请求速率

场景	消费者 1 向生产者 1 /request ⁻¹	消费者 1 向生产者 2 /request ⁻¹	消费者 2 向生产者 1 /request ⁻¹	消费者 2 向生产者 2 /request ⁻¹	消费者 3 向生产者 1 /request ⁻¹	消费者 3 向生产者 2 /request ⁻¹
场景 1	253	334	238	249	368	467
场景 2	240	319	155	334	430	372
场景 3	231	337	141	91	176	401
场景 4	132	213	133	204	244	190

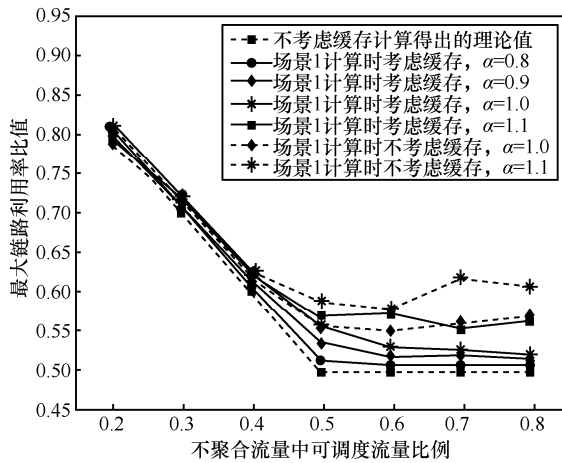


图2 场景1不同 Zipf 分布 α 值下最大链路利用率比值

用率相对默认情况下的比值降低幅度也不会有太大变化。值得注意的是，当 α 值为 1.0 和 1.1 时，缓存起的作用相对明显，计算时不考虑缓存因素相对考虑缓存因素的仿真实验结果较差且不稳定。

2) 与其他方法对比

作为对比，本文考察 Detti 等^[19]建模分析的 2 种多路径转发策略 UG 和 CF，全部流量为不聚合流量时， α 值取 0.9，针对场景 1，取不聚合流量中可调度流量比例为 0.2、0.3、0.4、0.5、0.6、0.7、0.8，计算相应的多路径流量分配结果并仿真，统计其最大链路利用率，分别与相同参数下全部流量为不可调度流量即不聚合流量中可调度流量比例为 0 时的默认情况下最大链路利用率取比值，对比本文方法的结果如图 3 所示。

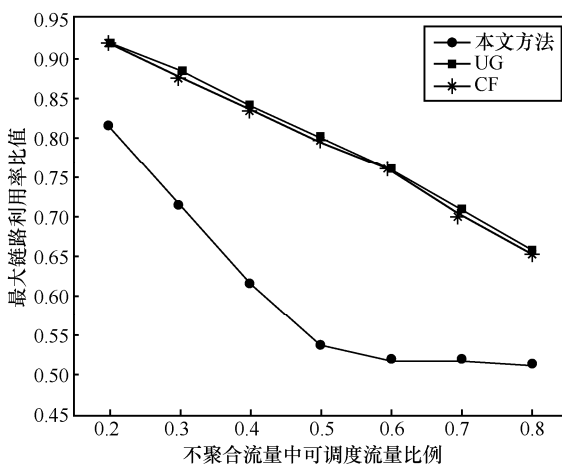


图3 $\alpha=0.9$ 时，场景 1 下采用不同方法得到的最大链路利用率比值

由图 3 可知，本文方法相比 UG、CF 来说，流量调度效果较好，这主要是因为 UG、CF 虽然也是典型的多路径转发策略，却无全局优化，故难以在多个消

费和多个生产者的场景下取得最理想的结果。

3) 不同场景下的效果和优化代价评估

当全部流量为不聚合流量时， α 值取 0.9，针对场景 1~场景 4，取不聚合流量中可调度流量比例为 0.2、0.3、0.4、0.5、0.6、0.7、0.8，按上述条件计算结果并仿真，统计其最大链路利用率，分别与相同参数下全部流量为不可调度流量即不聚合流量中可调度流量比例为 0 时的默认情况下最大链路利用率取比值，结果如图 4 所示。

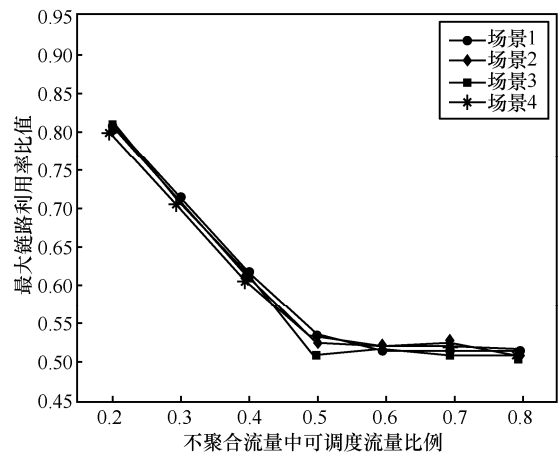


图4 $\alpha=0.9$ 时，不同场景下最大链路利用率比值

由图 4 可知，在这 4 种场景下，本文方法都能获得较好的调度效果，与之对应的路径长度拉伸的程度分别根据仿真结果统计实测值及式(17)求理论值，如图 5 所示。

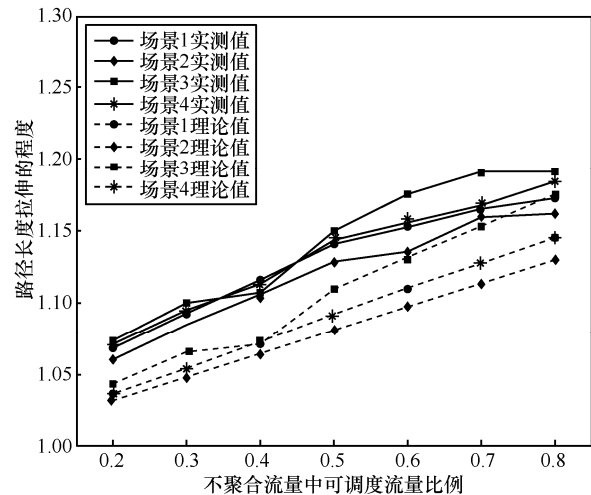


图5 $\alpha=0.9$ 时，不同场景下路径长度拉伸的程度

由图 5 可知，通常情况下，可调度流量比例越大，其代价即路径长度拉伸的程度越高；尽管相同情况下的实测值比理论值大 0.01~0.06，实测值和理

论值随着可调度流量比例变化的趋势基本一致。因此如需优化代价尽量小，只需调度不聚合流量中 50%~60% 的流量，即可在代价相对较小的前提下取得较为理想的结果。

4) 有流量聚合时不同场景下的效果

当部分流量为可聚合不可调度流量时， α 取值为 0.9，针对场景 1~场景 4，令各消费者到各生产者的可聚合不可调度流量请求速率为表 3 中各场景下内容请求速率总和的 5%，分别为 95、92、69、56 个 Interest 包请求/秒，则可聚合不可调度流量请求总和占有所有流量请求总和的 30%，各消费者到各生产者的各类流量请求速率之和仍依照表 3，取不聚合流量中可调度流量比例为 0.2、0.3、0.4、0.5、0.6、0.7、0.8，按上述条件计算结果并仿真，统计其最大链路利用率，分别与相同参数下全部不聚合流量为不可调度流量即不聚合流量中可调度流量比例为 0 时的默认情况下最大链路利用率取值，结果如图 6 所示。

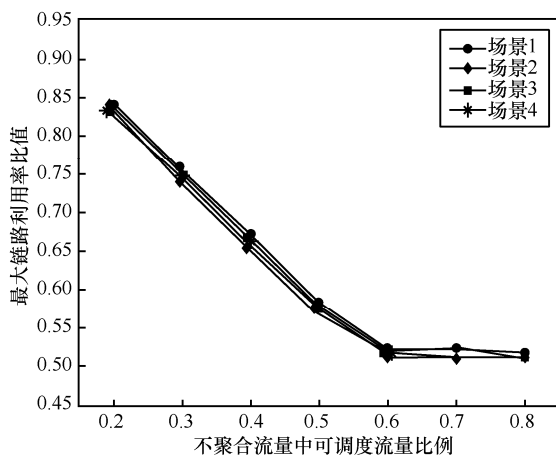


图 6 $\alpha=0.9$ 且有聚合流量时，不同场景下最大链路利用率比值

由图 6 可知，此时的结果与不可聚合不可调度流量时的结果相似，不过受可聚合不可调度流量影响，不聚合流量中可调度流量比例需达到 0.6 左右时，最大链路利用率相对默认情况下的比值才接近最优。

5) 控制器与节点间的通信开销

先取可调度流量比例为 0.2、0.3、0.4、0.5、0.6、0.7、0.8，改变 Zipf 分布的 α 值，分别取 0.8、0.9、1.0、1.1，根据 4.2.5 节的相关分析和式(18)、式(19)，当内容前缀总数 $A=1\ 000$ 个且优先调度热门内容的流量需求时，可计算出控制器与节点间除周期性信息采集外的通信开销最小值，以需调度的前缀数量来衡量，结果如图 7 所示；然后改变 A 值，取可调

度流量比例为 0.5、0.6，改变 α 值，分别取 0.8、0.9、1.0、1.1，结果如图 8 所示。

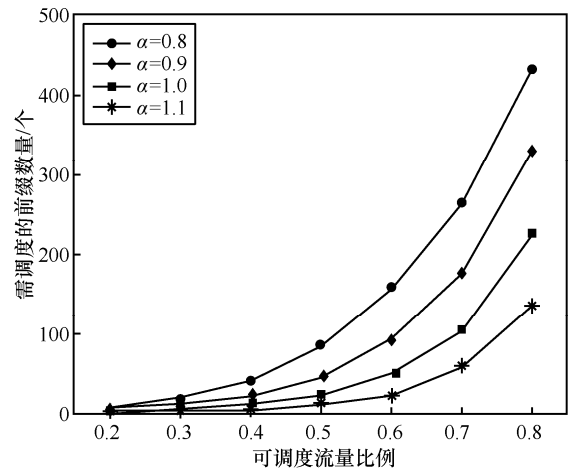


图 7 $A=1\ 000$ 个时控制器与节点间的通信开销最小值

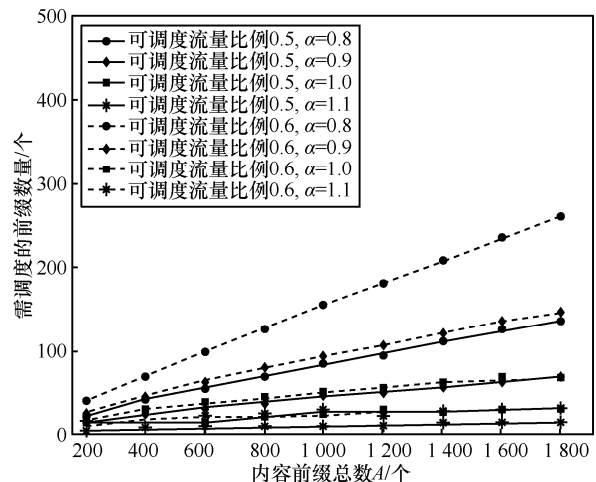


图 8 不同 A 值时控制器与节点间的通信开销最小值

由图 7 可知，当可调度流量比例相同时， α 越大，流量需求越集中于少数内容前缀，通信开销最小值就越小；若可调度流量比例增加，通信开销最小值以更快的速度增加。由图 8 可知，内容前缀总数增加也会使通信开销最小值随之增加；若只调度 50%~60% 的流量，通信开销最小值相对内容前缀总数的增长速度会缓慢得多，只是略有增加而已。

总之，根据实验仿真结果可知，当不聚合流量中可调度流量比例为 0.5~0.6 时，可在只付出较小的优化代价和略微增加控制器与节点间通信开销的前提下，将最大链路利用率相对默认情况下的比值降低 40% 以上。

以上仿真实验表明，本文方法相对默认情况和 UG、CF 等典型多路径转发策略能有效降低最大链路利用率，从而起到了均衡负载、提高网络吞吐量

的作用; 另一方面, 也验证了本文所做的假设和设计的合理性, 即只需在部分指定节点针对热门内容的流量需求部署合适的转发策略即可获得较好的效果, 且优化代价较小, 通信开销仅略有增加。

6 结束语

本文针对命名数据网络中流量的优化调度需求, 提出将 NDN 与 SDN 相结合以解决此问题, 设计整体架构和各类设备的功能。通过建模分析, 考虑 NDN 中 Interest 包的作用和特点及网内缓存和 Interest 包聚合因素, 提出一种可行的算法, 在部分节点针对热门内容的流量需求部署合适的转发策略, 进行全局性优化调度, 此外还评估了优化代价, 分析了控制器与节点间的通信开销。实验结果表明, 在指定节点针对热门内容的流量需求依照本文方法进行调度, 能在不产生大的优化代价及通信开销的前提下显著提高网络性能。

参考文献:

- [1] JACOBSON V, SMETTERS D K, THORNTON J D, et al. Networking named content[C]//The 5th International Conference on Emerging Networking Experiments and Technologies. 2009: 1-12.
- [2] PASSARELLA A. A survey on content-centric technologies for the current Internet: CDN and P2P solutions[J]. *Computer Communications*, 2012, 35(1): 1-32.
- [3] RAVINDRAN R, CHAKRABORTI A, AMIN S O, et al. 5G-ICN: delivering ICN services over 5G using network slicing[J]. *IEEE Communications Magazine*, 2017, 55(5): 101-107.
- [4] AHLGREN B, DANNEWITZ C, IMBRENDA C, et al. A survey of information-centric networking[J]. *IEEE Communications Magazine*, 2012, 50(7): 26-36.
- [5] ZHANG L, AFANASYEV A, BURKE J, et al. Named data networking[J]. *ACM SIGCOMM Computer Communication Review*, 2014, 44(3): 66-73.
- [6] YI C, AFANASYEV A, WANG L, et al. Adaptive forwarding in named data networking[J]. *ACM SIGCOMM Computer Communication Review*, 2012, 42(3): 62-67.
- [7] HOQUE A K M, AMIN S O, ALYYAN A, et al. NLSR: named-data link state routing protocol[C]//The 3rd ACM SIGCOMM Workshop on Information-Centric Networking. 2013: 15-20.
- [8] LEHMAN V, GAWANDE A, ZHANG B, et al. An experimental investigation of hyperbolic routing with a smart forwarding plane in NDN[C]//The 24th IEEE/ACM International Symposium on Quality of Service. 2016: 1-10.
- [9] KREUTZ D, RAMOS F M V, VERISSIMO P E, et al. Software-defined networking: a comprehensive survey[J]. *Proceedings of the IEEE*, 2015, 103(1): 14-76.
- [10] TORRES J, FERRAZ L, DUARTE O. Controller-based routing scheme for named data network[J]. *Electrical Engineering Program, COPPE/UFRJ Tech Rep*, 2012: 1-6.
- [11] CHANDA A, WESTPHAL C, RAYCHAUDHURI D. Content based traffic engineering in software defined information centric networks[C]//IEEE Conference on Computer Communications Workshops. 2013: 357-362.
- [12] BACHER F, RAINER B, HELWAGNER H. Towards controller-aided multimedia dissemination in named data networking[C]//IEEE International Conference on Multimedia & Expo Workshops. 2015: 1-6.
- [13] GAO S, ZENG Y, LUO H, et al. Scalable control plane for intra-domain communication in software defined information centric networking[J]. *Future Generation Computer Systems*, 2016, 56: 110-120.
- [14] SALSANO S, BLEFARI-MELAZZI N, DETTI A, et al. Information centric networking over SDN and OpenFlow: architectural aspects and experiments on the OFELIA testbed[J]. *Computer Networks*, 2013, 57(16): 3207-3221.
- [15] VAN A N L M, KUIPERS F A. NDNFlow: software-defined named data networking[C]//IEEE Conference on Network Softwarization. 2015: 1-5.
- [16] MAHMOOD A, CASETTI C, CHIASSERINI C F, et al. Efficient caching through stateful SDN in named data networking[J]. *Transactions on Emerging Telecommunications Technologies*, 2018, 29(1): 1-21.
- [17] POSCH D, RAINER B, HELWAGNER H. Saf: stochastic adaptive forwarding in named data networking[J]. *IEEE/ACM Transactions on Networking*, 2017, 25(2): 1089-1102.
- [18] CAROFIGLIO G, GALLO M, MUSCARIELLO L. Optimal multipath congestion control and request forwarding in information-centric networks: protocol design and experimentation[J]. *Computer Networks*, 2016, 110: 104-117.
- [19] DETTI A, PISA C, MELAZZI N B. Modeling multipath forwarding strategies in information centric networks[C]//IEEE Conference on Computer Communications Workshops. 2015: 324-329.
- [20] XIN Y, LI Y, WANG W, et al. Content aware multi-path forwarding strategy in information centric networking[C]//IEEE Symposium on Computers and Communications. 2016: 816-823.
- [21] UDUGAMA A, ZHANG X, KULADINITHI K, et al. An on-demand multi-path interest forwarding strategy for content retrievals in CCN[C]//IEEE/IFIP Network Operations and Management Symposium. 2014: 1-6.
- [22] KERROUCHE A, SENOUCI M R, MELLOUK A. QoS-FS: a new forwarding strategy with QoS for routing in named data network-

- ing[C]//IEEE International Conference on Communications. 2016: 1-7.
- [23] GARG N, KOENEMANN J. Faster and simpler algorithms for multicommodity flow and other fractional packing problems[J]. SIAM Journal on Computing, 2007, 37(2): 630-652.
- [24] BRESLAU L, CAO P, FAN L, et al. Web caching and Zipf-like distributions: evidence and implications[C]//The Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. 1999: 126-134.
- [25] ENNS R, BJORKLUND M, SCHOENWALDER J. Network configuration protocol(NETCONF)[J]. IETF RFC 6241, 2011: 1-113.
- [26] CHARALAMBOUS C, CONN A R. An efficient method to solve the minimax problem directly[J]. SIAM Journal on Numerical Analysis, 1978, 15(1): 162-187.
- [27] YEN J Y. Finding the k shortest loopless paths in a network[J]. Management Science, 1971, 17(11): 712-716.
- [28] 智江, 李俊, 吴海博, 等. 基于边缘优先的 ICN 缓存协作策略[J]. 通信学报, 2017, 38(3): 53-64.
ZHI J, LI J, WU H B, et al. Edge-first-based cooperative caching strategy in information centric networking[J]. Journal on Communications, 2017, 38(3): 53-64.
- [29] 唐明董, 张国清, 杨景, 等. 互联网可扩展路由[J]. 软件学报, 2010, 21(10): 2524-2541.
TANG M D, ZHANG G Q, YANG J, et al. Scalable routing for the Internet[J]. Journal of Software, 2010, 21(10): 2524-2541.
- [30] HENDERSON T R, LACAGE M, RILEY G F, et al. Network simulations with the NS-3 simulator[J]. ACM SIGCOMM Demonstration, 2008, 14(14): 527.
- [31] MASTORAKIS S, AFANASYEV A, ZHANG L. On the evolution of NDN-SIM: an open-source simulator for NDN experimentation[J]. ACM SIGCOMM Computer Communication Review, 2017, 47(3): 19-33.
- [32] UHLIG S, QUOTIN B, LEPROPRE J, et al. Providing public intradomain traffic matrices to the research community[J]. ACM SIGCOMM Computer Communication Review, 2006, 36(1): 83-86.
- [33] WANG Y, LI Z, TYSON G, et al. Optimal cache allocation for content-centric networking[C]//The 21st IEEE International Conference on Network Protocols. 2013: 1-10.

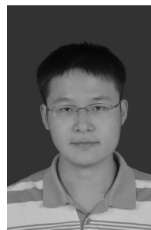
[作者简介]



董谦 (1986-), 男, 湖北咸宁人, 中国科学院计算机网络信息中心博士生, 佛山科学技术学院讲师, 主要研究方向为未来互联网、软件定义网络、流量工程等。



李俊 (1968-), 男, 安徽桐城人, 博士, 中国科学院计算机网络信息中心研究员、副总工程师、博士生导师, 主要研究方向为未来互联网、网络安全等。



马宇翔 (1991-), 男, 河南开封人, 中国科学院计算机网络信息中心博士生, 主要研究方向为网络体系结构、网络安全等。