



命名数据网络中 VIP 转发方案的性能优化

金 洋,吕金阳,谭小彬

(中国科学技术大学 未来网络实验室,合肥 230027)

摘 要:命名数据网络(named data networking, NDN)是信息中心网络(information centric network, ICN)的一个典型实例,是未来网络一种可能的架构。为优化 NDN 的转发性能,要求转发平面能根据不同的网络环境自适应地调整转发策略、选出最优的转发接口。“虚拟兴趣分组”(virtual interest packet, VIP)架构是一种基于 NDN 的联合动态转发与缓存的网络架构,通过 VIP 架构对转发方案进行了性能优化。VIP 架构的转发方案使用了背压算法,该算法在网络处于轻负载状态时时延较大,为此引入偏差函数进行优化。为检测优化效果,对原始 VIP 架构和优化后的 VIP 架构的转发性能进行了仿真实验对比。实验结果显示,优化后的转发方案相比原方案在转发时延上有一定降低。且由于偏差函数的引入不会增加原转发方案的时间复杂度,因此提出的方法可以优化 VIP 转发方案的性能。

关键词:命名数据网络;转发策略;背压算法;性能优化

中图分类号:TP393

文献标志码:A

文章编号:1673-825X(2018)01-0075-07

Performance optimization of VIP forwarding scheme in named data networking

JIN Yang, LV Jinyang, TAN Xiaobin

(Laboratory for Future Networks, University of Science and Technology of China, Hefei 230027 P. R. China)

Abstract: Named data networking (NDN) is a typical instance of information-centric networking (ICN), which is a possible framework for future network. To optimize the forwarding performance of NDN, it is necessary for forwarding plane to adjust forwarding strategies and choose the most suitable forwarding interfaces adaptively. The “virtual interest packet” (VIP) framework, within the context on the NDN, is designed for distributed dynamic forwarding and dynamic caching. The performance of forwarding scheme based on the VIP framework is optimized in this paper. The VIP framework employs a backpressure algorithm for forwarding. When the network is lightly loaded, there will be a larger delay. A bias function is proposed to improve the forwarding performance. Simulation experiments are designed to check the forwarding performance of VIP framework and the optimized one. The experimental result demonstrates the superior delay performance of the optimized forwarding scheme. Furthermore, the bias function doesn't increase the time complexity of the original scheme, which implies the method proposed in this paper can optimize the performance of VIP forwarding scheme.

Keywords: named data networking; forwarding strategy; backpressure algorithm; performance optimization

收稿日期:2017-09-25 修订日期:2017-12-18 通讯作者:金 洋 yjinking@mail.ustc.edu.cn

基金项目:国家自然科学基金(61673360,61233003);中兴通讯产学研合作基金

Foundation Items: The National Natural Science Foundation of China (61673360, 61233003); The ZTE Industry-Academia-Research Cooperation Funds

0 引言

网络架构需要与用户需求相匹配。当前互联网用户的主要需求已从主机间的通信转变为对网络内容的重复访问,传统的 TCP/IP 协议架构在拓展性、移动性、安全性等方面显示出诸多不适应性。信息中心网络试图从根本上解决这些问题^[1]。命名数据网络作为信息中心网络的实例之一^[2],凭借先进的理念、方案的可行性和诸多实质性进展,目前已经成为信息中心网络的主流方案。

NDN 不同于 IP 网络的会话模型,而是采用一种用户驱动、名字寻址的通信方式,实现了从“Where”到“What”,“Push”到“Pull”的转化,更符合用户关注内容本身的需求。NDN 每个节点配有内容缓存表(content store, CS)、待定兴趣表(pending interest table, PIT)、转发表(forwarding information base, FIB)3 种数据结构,通过传输兴趣分组(interest packet)、数据分组(data packet)完成内容请求和获取。分组的分装方式、寻址方式相对 TCP/IP 网络有了根本的不同。缓存的存在减少了相同数据的重复传输,提高了网络传输性能。NDN 在网络可扩展性、移动性、安全性、多路传输上有很大优势。

路由和转发是网络协议的核心部分,优秀的路由、转发算法有利于提高网络的传输性能。TCP/IP 网络的转发平面仅根据转发表作出转发决策,不支持多路传输,适应性较差。NDN 的转发表中,一个信息名称可以对应着多个输出端口,这为多播传输提供了直接支持,若选择从所有的端口进行转发,无疑增加了冗余数据的传输,容易造成网络拥塞。因此,若转发算法能根据网络状态的变化调整转发端口,使得转发平面具有自适应性,这将对网络性能的提高有重要作用^[3]。

在 NDN 的转发算法研究中,VIP 架构是一种联合动态转发与缓存的网络架构^[4],转发平面能根据网络状态的变化调整转发端口选择、缓存内容的替换,实现了网络吞吐量的最大化。但是当网络处于轻负载状态时,该架构将造成很大的传输时延,针对该问题,本文给出了一种解决方案。本文主要关注 VIP 架构的转发方案。

1 相关工作

随着 NDN 项目的进行,NDN 转发算法的研究已取得一定成果。根据当前转发策略所感知的信息类别,其大致可划分为以下三类:①基于网络端口状

况感知的转发算法;②基于网络内容分布感知的转发算法;③基于网络端口状况和内容分布共同感知的转发算法。

1.1 基于网络端口状况感知的转发算法

依据 NDN 转发平面所带来的网络状态信息,文献[3]提出一种基于端口着色的转发策略。在该策略中,红、黄、绿三色之一被标记在网络各个端口中,它们分别代表了 3 种不同的工作状态。在实际的转发过程中,端口颜色依据不同事件的发生而相互转换。在其转发决策中,绿色端口优先被选择,黄色端口只有在没有绿色端口的情况下才被用于转发,红色端口不能转发数据。通过对该策略分析容易发现:当有两份相同内容被缓存在网络节点中,一份离用户很近,另一份离用户很远,根据当前的转发策略,用户可能会从较远的源节点获取数据。换言之,该转发策略不具有网络内容分布感知能力。

Posch 等^[5]提出一种随机自适应的转发策略。该策略通过模仿水管系统自调节方法,智能指导和分发兴趣分组,以绕过由链路故障、拥塞等导致的瓶颈节点。通过设置信任阈值,网络所有转发端口被分为信任端口与不信任端口。在实际转发过程中,每过一段时间,各个端口都会被评定一次。信任端口的转发概率大于不信任端口的转发概率。通过对该转发策略的工作原理分析,我们发现该转发策略只适合于高度拥塞的场景,对于普通的场景,由于丢包率不大,很难得到一个很好的转发效果。与此同时,该策略也不具有网络内容分布的感知能力。

1.2 基于网络内容分布感知的转发策略

Yeh 等^[4]在原有命名数据网络的基础上,提出一种联合动态转发与缓存的网络架构。在该网络架构中,数据的转发被分为 2 个层面来进行,即虚拟的控制平面和实际的转发平面。在虚拟控制平面中,基于用户需求度量的虚拟兴趣包被转发。而实际平面中,实际的兴趣包和数据包被处理。为了实现期望的目标,虚拟控制平面通过分布式算法操作虚拟兴趣包,而产生流率和队列长度被用于指导现实平面中兴趣包的转发以及数据包的缓存。

Rossini 等认为 NDN 缓存与转发之间存在一种紧密的耦合性,并提出一种最近邻转发策略^[6]。该策略在其实现过程中融入一种集中控制思想,具体部署如下:每个路由节点都能获取完整的网络拓扑结构以及所有路由节点的缓存信息,然后通过一种扩散的方式找到最近的缓存源,最后再向该源节点

转发。该转发策略并没有对网络的实时状态进行感知,同时该策略随着网络拓扑变大和网络中转发数据量的增加而变得难以实现。此外,文献[7]实现了一种基于势能的转发策略,文献[8]介绍了一种基于热度的拥塞控制策略,文献[9]提出一种通过区分不同服务的生命时间来提高 QoS 的策略。

1.3 基于网络端口状况和内容分布共同感知的转发策略

文献[10]实现了一种按需分配的多路径转发策略。在转发处理流程中,端口时延大小不仅被作为该端口状况的衡量标准,以用于端口的概率计算,而且以一种平滑的方式被实时更新。尽管该策略通过不同端口下具体内容延迟度量的方式实现了一种网络端口状况和内容分布的共同感知,但其实际的感知效果较差,究其根本,该策略感知的对象是具体内容前缀,其层次太低。尽管在文献[11]中,Gao 等针对以上问题提出一种改善方案,即提高网络状态的实时性,但这种方案并没有从根本上解决上述问题。

2 VIP 转发方案和分析

VIP 是虚拟兴趣分组(virtual interest packet)的缩写。VIP 架构是一种基于 NDN 的分布式动态转发和动态缓存的架构。VIP 计数器记录了局部区域内对每个数据对象的需求度。如图 1,该架构在实际的转发平面以外构造了一个虚拟平面(该平面上并不转发实际的兴趣分组和数据分组,只转发 VIP)。通过虚拟平面计算出的转发端口、传输速率来指导实际平面进行转发,VIP 架构实现了网络吞吐量的最大化,且设计了高效的内容缓存方案。

2.1 VIP 架构的关键组成部分及变量

2.1.1 虚拟兴趣分组

VIP 架构构造了一种特殊的分组——虚拟兴趣分组,用户一旦请求某个数据对象,一个对应的 VIP 便产生了。虚拟平面上虚拟兴趣分组的转发与实际平面上的兴趣分组的转发存在以下 2 点不同。

1) 粒度。如图 1, VIP 对应的是一个 Data Object,相比 Interest Packet 对应的 Data Packet,粒度更大。但由于 VIP 和 Interest Packet 内主要包含所请求对象/块的名称,两者的大小几乎相同。VIP 的这种设计可以使得虚拟平面上的算法复杂度大大降低。

2) 转发机制。虚拟平面上不设置 PIT,因此, VIP 被转发后只有在到达请求数据对象的缓存路由

器或内容源时,才会停止转发,到达这些地方后, VIP 被丢弃。在实际转发平面上,一个 Interest Packet 到达一个路由器节点,若 PIT 中已存有对相同数据分组的表项,Interest Packet 便停止转发。

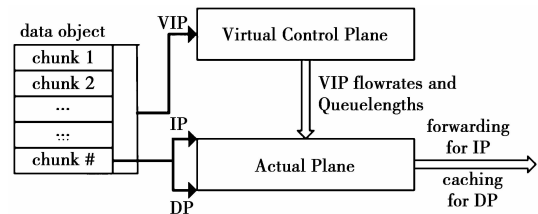


图 1 VIP 架构(IP 代表 Interest Packets, DP 代表 Data Packets)

Fig. 1 VIP framework (IP stands for Interest Packets, DP stands for Data Packets)

2.1.2 VIP 计数器

虚拟平面上每个节点还维持着对每个内容请求的 VIP 队列长度,每到达一个 VIP,对应的 VIP 队列长度便增加 1,这个队列长度我们使用一个 VIP 计数器来表示:在时刻 t , n 节点处请求内容 k 的虚拟兴趣包个数记为 $V_n^k(t)$ 。VIP 计数器的值代表了局部区域内对某数据对象的请求频繁度。

在内容 k 的缓存路由器中, VIP 计数器值较小。到达内容源 $src(k)$ 处,计数器值为 0,“水往低处流”是该模型一个很直观的比喻。因此,一个自然的思路便是,内容 k 的虚拟兴趣分组应转发至内容 k 的 VIP 计数器值越低的节点。结合文献[4],容易得到 VIP 计数器的状态转移方程。

2.1.3 虚拟平面

虚拟平面通过 VIP 的转发计算出转发端口、传输速率来指导实际平面进行转发。由于 VIP 架构是分布式、动态的,它需要与相邻节点交换参数,这一部分占用了实际带宽,但由于虚拟平面的操作都在 object 层面,这将导致要传输的参数大大减少(相比于 chunk)。相邻节点在实际传输参数时可以把多个参数的信息放进一个传输分组中。所以交换参数所占用的带宽并不会太大。

2.2 VIP 转发方案

VIP 架构的虚平面上的转发方案采用了背压算法^[12]。背压算法根据相邻两点的积压差(在本文中即为相邻节点同一内容 VIP 计数器值的差)来决定调度策略^[13],当有多个传输对象要通过一个信道时,背压算法优先让积压差大的内容进行传输。积压差越大,算法判断下一跳节点对该内容的传输能力更强,应优先向其传输该内容;积压差越小,算法

判断出下一跳节点的传输能力较弱,应减少向其传输内容,或寻找其他链路。

实际转发平面上我们引入一个时间窗口 T ,用于计算过去 T 时间内链路对虚拟兴趣分组的平均转发速率。当兴趣分组到达节点时,选择对该内容平均转发速率最大的链路进行转发。

2.3 VIP 转发方案不足分析

VIP 转发方案采用的背压算法能实现网络吞吐量最大化,但背压算法在网络处于轻负载状态时存在时延大的缺点。

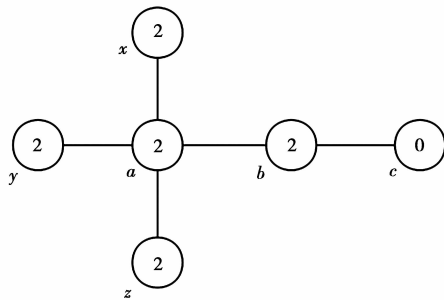


图 2 网络轻载状态下节点 VIP 计数器
Fig. 2 VIP counts of nodes when the network is lightly loaded

以图 2 为例,有节点 a, b, c, x, y, z , 假设此网络中传输的内容只有一种—— k , c 为内容 k 的存储源节点。节点 a 正在请求内容 k , 圆内的数字为当前各节点对内容 k 的 VIP 计数器值。采用虚拟平面上转发方案,那么和 a 相连的任一条链路上的积压差均为 0, 无法给实际平面提供有效的指导。最终实际平面上 a 的路由器将随机选择链路用于传输 k 的兴趣分组。

因此,在网络处于轻负载状态时,各点中的 VIP 计数器值小且节点间较为接近,甚至为零,没有明显的积压差,背压算法效果差甚至失效,当实际的网络规模很大时,易造成较大的转发时延,降低转发性能。

3 网络轻载时 VIP 转发方案优化

3.1 优化思路

VIP 转发方案计算积压差时,仅利用相邻的节点 VIP 计数器值来计算,即一跳内。图 2 中,节点 b 的邻居节点 c 即为内容源,如果在计算积压差时同时考虑一跳以外的节点信息^[12,14],使得朝向内容源方向的积压差更大,那么分组将会朝内容源方向传输,以此改善原 VIP 架构在网络轻载时时延较大的现象。

为实现以上目的,可以尝试对当前节点 VIP 计数器值加上一个偏差量,用于增大朝向内容源方向

的积压差。一般可把当前节点距离内容源的最小跳数作为偏差量。

仍以图 2 为例,点 a, b 距离内容源 c 的跳数分别为 2, 1; 点 x, y, z 距离点 c 均为 3 跳,调整后的 VIP 计数器值为

$$V_a^k(t)' = 2 + 2 = 4$$

$$V_b^k(t)' = 2 + 1 = 3$$

$$V_x^k(t)' = V_y^k(t)' = V_z^k(t)' = 2 + 3 = 5$$

再计算与 a 相连的各链路的积压差, ab 上达到最大(为 1, 其余均为 -1), 因此,点 a 的请求将沿链路 (a, b) 上传输。

3.2 偏差函数

基于 3.1 节的分析,我们引入偏差函数,偏差函数的作用就是调整 VIP 计数器的值,调整背压算法计算出的积压差结果,使得兴趣分组能够更准确地向内容源传输。设当前时隙为 t , 虚拟平面上 n 节点请求内容 k , 我们记对应于 VIP 计数器 $V_n^k(t)$ 的偏差函数为 $f_n^k(t)$, 那么修正后的 VIP 计数器为

$$V_n^k(t)' = V_n^k(t) + f_n^k(t) \quad (1)$$

偏差函数的形式为

$$f_n^k(t) = \eta_n^k(t) \text{dist}(n, \text{src}(k)) \quad (2)$$

(2) 式中: $\text{dist}(n, \text{src}(k))$ 为节点 n 与内容源间的最小跳数, $\eta_n^k(t)$ 为系数,使 $f_n^k(t)$ 比 $V_n^k(t)$ 低一个数量级或处于相同数量级。 $\eta_n^k(t)$ 对偏差函数数量级的控制可以保证积压差的计算在一般情况下仍由 VIP 队长主导,使得转发方案仍然是基于背压算法。

值得注意的是,文献[15]中同样采用了偏差函数来对原 VIP 架构转发方案作出改进,本文与其在优化的场景、偏差函数定义存在不同。

本文重点讨论网络轻负载场景。在此场景下,周围节点 VIP 队长小甚至为 0, 根据 2.3 节的分析,这将导致背压算法效果差甚至失效,因此,偏差函数是基于节点到内容源的最短路径,而不是基于 VIP 队长。文献[15]考虑的是一般网络场景,在原 VIP 架构基础上,其希望通过联合动态转发、缓存、拥塞控制机制进一步提升时延性能。文献[15]并不着重关注网络轻负载,因此,不考虑周围节点 VIP 队长小甚至为 0 的情况,背压算法仍能正常工作。在该场景中,节点 n 关于内容 k 的偏差函数可基于周围节点 VIP 队长^[16], 其中一种具体形式为

$$f_n^k(t) = \frac{1}{z} \min_{n' \in \{n'; (n, n') \in L^k\}} V_{n'}^k(t) \quad (3)$$

(3) 式中, z 为稳定网络的参数。但由于轻负载状态中缺乏足够的 VIP 队长信息,此形式的偏差函数并

不适用本文。

3.3 引入偏差函数后的转发方案

引入偏差函数后需要对积压差的计算做改变,即链路 (a, b) 上对内容 k 的积压差为

$$W_{ab}^k(t) = V_a^k(t)' - V_b^k(t)' \quad (4)$$

调整积压差的计算后,虚平面上的转发算法再按背压算法进行,再计算出 VIP 的最大传输速率和实际传输速率,即算法 3.1(其中, N 为节点集, K 为内容集, L 为链路集, L^k 为允许传输内容 k 的链路集)。

算法 3.1 中, $k_{ab}^*(t)$ 是根据最大积压差决定的在链路 (a, b) 上应传输的内容。链路 (a, b) 上允许 VIP 传输的最大速率 $\mu_{ab}^k(t)$ 是根据反向链路 (b, a) 的容量 C_{ba} 得出的,这是因为在反向链路上传输的数据分组大小远大于兴趣分组,这里设置兴趣分组的传输速度应和其接近可以防止拥塞。实际中 VIP 的数量可能很小,因此,需将 $\mu_{ab}^k(t)$ 做加权平均后得出实际传输速率 $v_{ab}^k(t)$ 。

实际平面的转发仍选择过去时间段内平均速率最大的链路转发。设 n 为当前节点,欲转发 k 的兴趣分组,则选择转发链路为 $\arg \max_{b:(n,b) \in L^k} \{v_{nb}^k(t)\}$,其中,

$$\bar{v}_{nb}^k(t) = \frac{1}{T} \sum_{t'=t-T+1}^t v_{nb}^k(t')$$

3.4 工作流程

虚拟平面工作流程如图 3 所示,虚拟平面先按照算法 3.1 完成转发速率的计算,再按照 VIP 状态转移方程更新 VIP 计数器值。

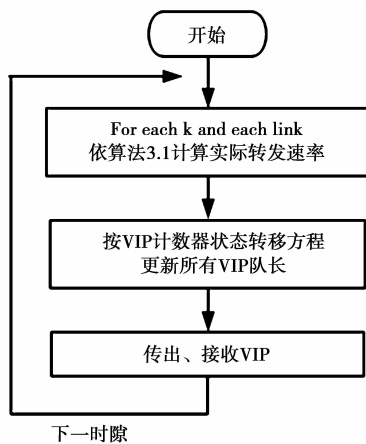


图 3 虚拟平面工作流程

Fig. 3 Flow chart of virtual control plane

算法 3.1 虚拟平面上的转发算法。

Input: $V_n^k(t) (\forall n \in N, \forall k \in K)$;

Output: $\mu_{ab}^k(t), v_{ab}^k(t) (\forall k \in K, \forall (a, b) \in L^k)$;

Step 1: For each link (a, b) do

1. 取 $\forall k_{ab}^*(t) \in \arg \max_{k:(a,b) \in L^k} \{W_{ab}^k(t)\}$;

2. $W_{ab}^*(t) = \max\{0, W_{ab}^{k_{ab}^*(t)}(t)\}$;

Step 2: For each k and each $(a, b) \in L^k$ do

$$\mu_{ab}^k(t) = \begin{cases} \frac{C_{ba}}{z}, W_{ab}^*(t) > 0 \text{ and } k = k_{ab}^*(t) \\ 0, \text{otherwise} \end{cases}$$

For $\forall (a, b) \notin L^k$ do $\mu_{ab}^k(t) = 0$;

Step 3: For each k and each $(a, b) \in L^k$ do

$$v_{ab}^k(t) = \min \left\{ \left[\frac{\mu_{ab}^k(t)}{\sum_{n:(a,n) \in L^k} \mu_{an}^k(t)} \cdot V_a^k(t) \right], \mu_{ab}^k(t) \right\};$$

For $\forall (a, b) \notin L^k$ do $v_{ab}^k(t) = 0$;

实际转发平面上,兴趣分组、数据分组的转发流程分别如图 4、图 5 所示。传输兴趣分组时,依次检查当前节点的 CS, PIT, 若命中则分别采取返回数据分组、PIT 增加接口的操作,并停止转发 Interest。当检查 FIB 时,采用虚拟平面计算的过去时间窗口内平均转发速率最高的链路进行转发。

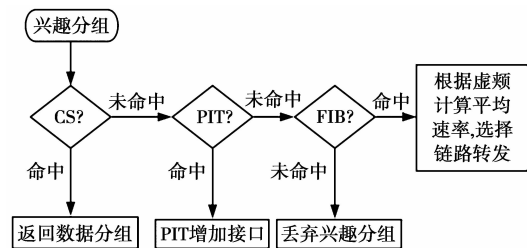


图 4 实际平面兴趣分组转发流程

Fig. 4 Flow chart of Interest forwarding in actual forwarding plane

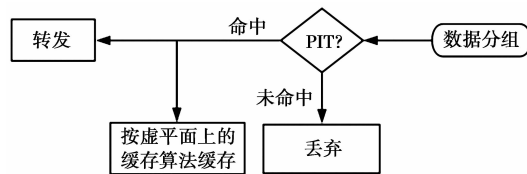


图 5 实际平面数据分组转发流程

Fig. 5 Flow chart of data forwarding in actual forwarding plane

当数据分组返回时,节点检查 PIT 决定是否转发,并依照缓存策略决定是否缓存和如何缓存。

3.5 算法复杂度对比

原 VIP 架构转发算法时间复杂度为 $O(|N|^2|K|)$,其中, $|N|, |K|$ 分别代表节点数目、内容对象数目^[4]。优化后的转发方案算法时间复杂度保持不变,理由如下。

优化的转发方案比原方案增加的工作量为①提

前计算节点与内容的最短距离(以跳数衡量);②计算 VIP 计数器采用 $V_n^k(t)' = V_n^k(t) + f_n^k(t)$ 。

针对①,需要计算每个内容源节点与所有节点的最短路径,可转化为解决 $|K|$ 次单源最短路径问题,算法复杂度为 $O(|N|^2|K|)$ 。并且本文假设网络拓扑不变,因此,最短距离可在初始时计算完毕,而不必在每次转发时计算。

针对②,偏差函数值的计算、VIP 计数器值的修改均在常数时间复杂度内完成。

综上,优化后的转发方案时间复杂度与原方案保持一致,均为 $O(|N|^2|K|)$ 。

4 仿真测试与分析

为对比原 VIP 架构和优化后的 VIP 架构的转发性能,我们设计了实验进行仿真。仿真平台为 Microsoft Visual Studio Community 2015,使用 C++ 语言。

实验采用了图 6 所示的 GEANT 网络拓扑,设置 22 个节点,不同内容数 300 种,并预先指定每个内容的存储位置,不同节点间最小距离跳数预先计算完毕。每个节点在每个时隙内按设定的请求速率随机发送兴趣分组。不考虑 PIT 定时器超时、兴趣分组重传。改进的转发方案 and 原 VIP 转发方案其余仿真参数保持一致,如表 1 所示,

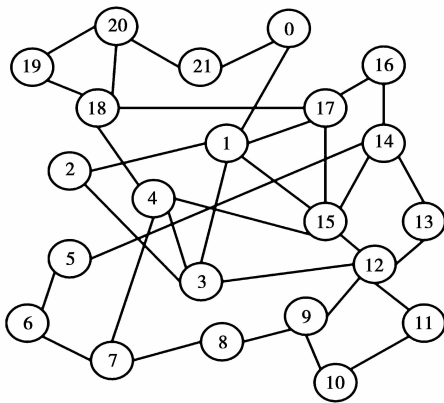


图 6 GEANT 网络拓扑

Fig. 6 GEANT network topology

表 1 仿真参数

Tab. 1 Simulation parameter

配置参数	参数取值
链路容量/(Mbit · time slot ⁻¹)	500
兴趣分组大小/B	125
数据分组大小/KB	50
数据对象大小/MB	5
数据对象数目	300
仿真时间/time slot	100
节点缓存容量/GB	2

我们定义时延为所有节点从发送一个兴趣分组直到接收到对应数据分组的历时总和。以时延作为转发性能指标。仿真时间为 100 个时隙,实验中调整每个节点发送兴趣分组的请求速率,记录总时延,每个请求速率下进行五次实验取平均值。

仿真结果如表 2、图 7 所示。在我们的仿真测试中,综合 6 种请求速率,改进后的转发方案在时延上比原 VIP 转发方案降低了 24%,当请求速率为 50 pkt/time slot 时,优化效果最明显,降低了 11 980 time slots(降低 40.4%)。

表 2 仿真结果

Tab. 2 Simulation results

请求速率 /(pkt · time slot ⁻¹)	原方案转发时延 /time slot	优化后转发时延 /time slot
10	5 112	4 166
20	11 871	12 773
30	12 040	10 160
40	20 432	13 282
50	29 681	17 701
60	24 080	20 320

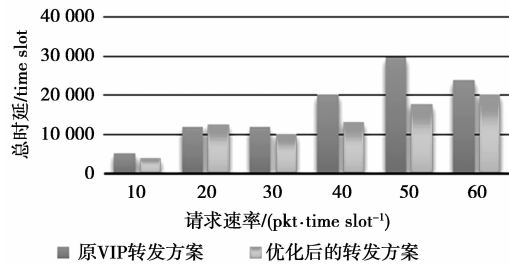


图 7 仿真时延对比图

Fig. 7 Simulation comparison chart of time delay

在请求速率为 20 pkt/time slot 时,原 VIP 转发方案比优化后的转发方案时延性能稍优,以下是可能的解释:GEANT 网络拓扑在规模上较小,每个节点的相邻节点数为 2—5,原背压算法在轻负载状态下虽然会随机转发,但仍有相当概率转发至更合适的链路——更快地到达内容缓存节点,而不必花费更长的时间转发至内容源节点。

实验结果表明,本文第 3 节优化思路是正确的,引入偏差函数后的转发方案可以达到时延性能优化的效果。

5 结束语

NDN 试图解决 TCP/IP 网络在当前环境下的诸多不适应性问题,但对于 NDN 本身,仍然在不少方面需要进行性能优化。VIP 架构具有良好的转发性能和缓存性能。VIP 架构在转发上使用了背压算

法,该算法在网络处于轻负载状态时,将导致传输时延较大。本文基于 VIP 架构的转发方案,引入偏差函数进行优化。仿真实验对优化前后的转发方案进行了实验对比。实验结果显示,优化后的 VIP 架构在时延上有一定降低。

在转发问题上,仍然存在诸多难题:如何对不同的内容实现不同的转发策略?查找时查询空间过大如何解决?网络中存在拥塞时如何恢复?这些问题的解决对 NDN 转发性能的提高有很大帮助。

除了使用软件进行实验仿真,文献[17]提到了使用一种 Banana Pi 的实际实验平台,使用 Banana Pi 进行实验,其结果将更为真实,这也是我们下一步的努力方向。

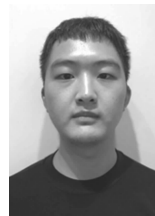
参考文献:

- [1] 吴超,张尧学,周悦芝,等. 信息中心网络发展研究综述[J]. 计算机学报, 2015, 38(3):455-471.
WU Chao,ZHANG Yaoyue,ZHOU Yuezhi, et al. A Survey for the Development of Information-Centric Networking [J]. Chinese Journal of Computers, 2015, 38(3):455-471.
- [2] ZHANG L, AFANASYEV A, BURKE J, et al. Named Data Networking[J]. Acm Sigcomm Computer Communication Review, 2014, 44(3):66-73.
- [3] CUI Ying, AFANASYEV A, MOISEENKO I, et al. A Case for Stateful Forwarding Plane[J]. Computer Communications, 2013, 36(7):779-791.
- [4] YEH E, HO T, CUI Ying, et al. VIP: A Framework for Joint Dynamic Forwarding and Caching in Named Data Networks[J]. Computer Science, 2014(28):117-126.
- [5] POSCH D,RAINER B,HELLWAGNER H. SAF:Stochastic Adaptive Forwarding in Named Data Networking [J]. IEEE/ACM Transactions on Networking,2015,PP(99):1-14.
- [6] ROSSINI G, ROSSI D. Coupling caching and forwarding: Benefits, analysis, and implementation[C]//Proceedings of the 1st international conference on Information-centric networking. New York: ACM, 2014: 127-136.
- [7] EUM S, NAKAUCHI K, MURATA M, et al. CATT:potential based routing with content caching for ICN[C]//Edition of the Icn Workshop on Information-Centric NETWORKING. Helsinki:ACM,2012:49-54.
- [8] PARK H, JANG H, KWON T. Popularity-based congestion control in named data networking[C]// International Conf on Ubiquitous & Future Networks. Shanghai: IEEE, 2014:166-171.
- [9] WU Tinyu, LEE W T, DUAN C Y, et al. Data Lifetime Enhancement for Improving QoS in NDN [J]. Procedia Computer Science, 2014(32):69-76.
- [10] UDUGAMA A, ZHANG XINYI, KULADINITHI K, et al. An On-demand Multi-Path Interest Forwarding strategy for content retrievals in CCN[C]// NOMS 2014-2014 IEEE/IFIP Network Operations and Management Symposium. New York: IEEE, 2014:1-6.
- [11] CAO Jianxun, PEI Dan, WU Zhelun, et al. Improving the freshness of NDN forwarding states [C]// Ifip NETWORKING Conference. Krakow, Vienna: IEEE Computer Society, 2016:189-197.
- [12] NEELY M J, URGAKONKAR R. Optimal Backpressure Routing for Wireless Networks with Multi-Receiver Diversity[C]//Information Sciences and Systems, 2006, Conference on. Princeton: IEEE, 2007:18-25.
- [13] YING Lei, SHAKKOTTAI S, REDDY A, et al. On Combining Shortest-Path and Back-Pressure Routing Over Multihop Wireless Networks [J]. IEEE/ACM Transactions on Networking, 2009, 19(3):841-854.
- [14] NEELY M J. Dynamic Power Allocation and Routing for Satellite and Wireless Networks with Time Varying Channels [J]. Massachusetts Institute of Technology, 2003 (31):231-241.
- [15] CUI Ying, LAI Fan, YEH E, et al. Enhanced VIP Algorithms for Forwarding, Caching, and Congestion Control in Named Data Networks [C]//Global communications conference. CA, USA: IEEE, 2017:1-7.
- [16] CUI Ying, YEH E, LIU Ran. Enhancing the delay performance of dynamic backpressure algorithms[C]// Signals, Systems and Computers, 2013 Asilomar Conference on. Pacific Grove: IEEE, 2013:27-31.
- [17] RAINER B, POSCH D, LEIBETSEDER A, et al. A Low-Cost NDN Testbed on Banana Pi Routers[J]. IEEE Communications Magazine, 2016, 54(9):105-111.

作者简介:



金 洋(1994—),男,浙江绍兴人,硕士研究生,主要研究方向为命名数据网络的转发性能优化。E-mail: yjinking@mail.ustc.edu.cn。



吕金阳(1993—),男,河南周口人,硕士研究生,主要研究方向为命名数据网络的转发性能优化。E-mail: jinyangl@mail.ustc.edu.cn。



谭小彬(1973—),男,陕西西安人,副教授,博士,主要研究方向为网络性能优化和信息安全。E-mail: xbtan@ustc.edu.cn。

(编辑:张 诚)