基于深度布隆过滤器的NDN网络三级名字查找方法

吴庆涛[①] 师君如[①] 张明川*[①] 王倩玉[①] 朱军龙[①] 张宏科[②]

[①](河南科技大学信息工程学院 洛阳 471023)

[②](北京交通大学下一代互联网互联设备国家工程实验室 北京 100044)

摘 要：为提高命名数据网络(Name Data Networking, NDN)路由过程中内容名字查找的效率，该文提出一种基于深度布隆过滤器的3级名字查找方法。该方法使用长短记忆神经网络(Long Short Term Memory, LSTM)与标准布隆过滤器相结合的方法优化名字查找过程；采用3级结构优化内容名字在内容存储器(Content Store, CS)、待定请求表(Pending Interest Table, PIT)中的精确查找过程，提高查找精度并降低内存消耗。从理论上分析了3级名字查找方法的假阳性率，并通过实验验证了该方法能够有效节省内存、降低查找过程的假阳性。

关键词：命名数据网络；内容名字查找；深度布隆过滤器；内存消耗

中图分类号：TN919.2; TP393 **文献标识码**：A **文章编号**：1009-5896(2021)12-3597-08

DOI: 10.11999/JEIT200766

A Three-level Name Lookup Method Based on Deep Bloom Filter for Named Data Networking

WU Qingtao[①] SHI Junru[①] ZHANG Mingchuan[①] WANG Qianyu[①]
ZHU Junlong[①] ZHANG Hongke[②]

[①](*School of Information Engineering, Henan University of Science and Technology, Luoyang* 471023, *China*)

[②](*National Engineering Laboratory for Next Generation Internet Interconnection Devices, Beijing Jiaotong University, Beijing* 100044, *China*)

Abstract: A three-level name lookup method based on deep Bloom filter is proposed to improve the searching efficiency of content name in the routing progress of the Named Data Networking (NDN). Firstly, in this method, the Long Short Term Memory (LSTM) is combined with standard Bloom filter to optimize the name searching progress. Secondly, a three-level structure is adopted to optimize the accurate content name lookup progresses in the Content Store (CS) and the Pending Interest Table (PIT) to promote lookup accuracy and reduce memory consumption. Finally, the error rate generated by content name searching method based on deep Bloom filter structure is analyzed in theory, and the experiment results prove that the proposed the three-level lookup structure can compress memory and decrease the error effectively.

Key words: Named Data Networking (NDN); Content name lookup; Deep Bloom filter; Memory consumption

1 引言

随着互联网的不断发展，基于IP协议的网络细腰模型逐渐暴露出诸多弊端，端到端的通信方式已经不能满足用户日益增长的网络需求。针对当前网络可扩展性差、动态性不足等问题，研究者抛开现有网络结构的束缚，提出未来互联网体系结构[1–3]以解决当前IP网络所面临的问题。命名数据网络(Name Data Networking, NDN)[4]作为变革式网络架构的代表之一，成为学术界研究的热点。在NDN网络中，以基于内容的通信方式取代传统网络基于IP的通信方式，每个信息内容都有其独立的、采用层次化方式命名的名字。网络中的节点能够缓存内容，用户通过包含内容名的兴趣包获取所需信息[5]。内容名字查找作为NDN路由器的一个关键功能，对整个网络的路由效率至关重要[6]。因

收稿日期：2020-08-27；改回日期：2021-09-24；网络出版：2021-10-22
*通信作者： 张明川 zhang_mch@haust.edu.cn
基金项目：国家自然科学基金(61871430, 61976243)，中原科技创新领军人才(214200510012)，河南省教育厅基础研究专项(19zx010)，河南省教育厅重点科研项目(20A520011)

Foundation Items: The National Natural Science Foundation of China (61871430, 61976243), The Leading Talents of Science and Technology in the Central Plain of China (214200510012), The Basic Research Projects in the University of Henan Province (19zx010), The Key Project of the Education Department Henan Province (20A520011)

此，如何设计高效的名字查找结构是NDN网络的关键难题之一。

信息量的增加会造成路由表规模扩大，在大规模路由表中实现准确、低内存消耗的内容名字查找，成为NDN网络名字查找算法的核心。为了更好地提供服务，每个路由节点需要维护3个表：内容存储器(Content Store, CS)、待定请求表(Pending Interest Table, PIT)、转发信息表(Forwarding Information Base, FIB)[7]。当用户请求数据时，会向网络发出一个请求内容的兴趣包。兴趣包中包含所请求内容的名字，而这个名字需要依次在CS, PIT, FIB表中进行查找以获得请求数据或对应的转发端口，帮助用户获取所需的信息内容[8]。本文主要关注内容名字在NDN网络路由节点的CS和PIT中进行精确查找的方法。

为实现内容名字在大规模路由表中的精确查找，不仅需要降低路由表的内存消耗，还需要提高查找的精度与效率。目前，一些研究者围绕名字查找结构提出了几种精确查找算法，如字符查找树[9]、布隆过滤器[10]和哈希表[11]等。为了优化元素间的冲突和减少内存占用量，Kraska等人[12]提出了学习索引结构，探索使用机器学习方法替代传统索引结构。为了提高学习索引的效率，Mitzenmacher[13]提出了使用两个布隆过滤器将学习函数夹在其间的夹心索引结构。这些算法在查找精度和内存使用方面有不同的优势，但还没有很好地解决名字之间的冲突问题，并且为了提高查找精度还会牺牲一定的内存空间。

也有一些研究者采用几种传统查找算法相结合的方式提升查找效率。文献[14]提出了利用哈希表进行内容名字查找，将每个内容名字前缀通过哈希函数进行计算，得到其关键值并存储在路由节点的FIB中，以提高查找的效率。文献[15]提出了一种基于字符查找树的名字查找结构，将名字前缀通过分隔符分割为多个名字组件，利用树状结构实现快速查找。文献[16]使用布隆过滤器进行名字查找，通过哈希函数将内容名字进行编码，存储在一个定长的位数组中，进行名字查找时只需判断对应位数组上的数字是否全为1即可。这些方法虽然可以在一定程度上提升查找效率，但一般都会产生假阳性，牺牲了查找的精确性。

本文提出一种基于深度布隆过滤器的内容名字查找结构。它的基本方法是：首先，利用布隆过滤器对查找内容名字进行预过滤，快速筛选出符合要求的内容；然后，引入长短记忆神经网络(Long Short Term Memory, LSTM)[17–19]对筛选出的内容名字进行精确查找，判断内容名字是否为表中的内容；最后，再利用一个小型备份过滤器对漏报内容进行3重查找。由此，可以提高内容名字在路由表中查找的准确性，减少路由表项的内存消耗，提升路由表查找效率。

2 基于深度布隆过滤器的名字查找结构与算法

本节首先对基于深度布隆过滤器的名字查找结构进行介绍；然后，提出了相应的查找算法，对内容名字在NDN网络中的精确查找过程进行解析；最后，对深度查找结构产生的假阳性进行分析。

2.1 深度布隆过滤器查找结构

深度布隆过滤器的查找结构主要分为3级，包含内容名字的用户兴趣包通过3级的查找，可以增加查找的精确度并减少内存占用。第1级进行预过滤，第2级进行精确查找，第3级消除漏报，如图1所示。

第1级初始过滤器主要采用布隆过滤器对名字进行预处理，目的是判断所查找的名字是否为内容集合中的成员或近似成员，实现快速初步筛选。如

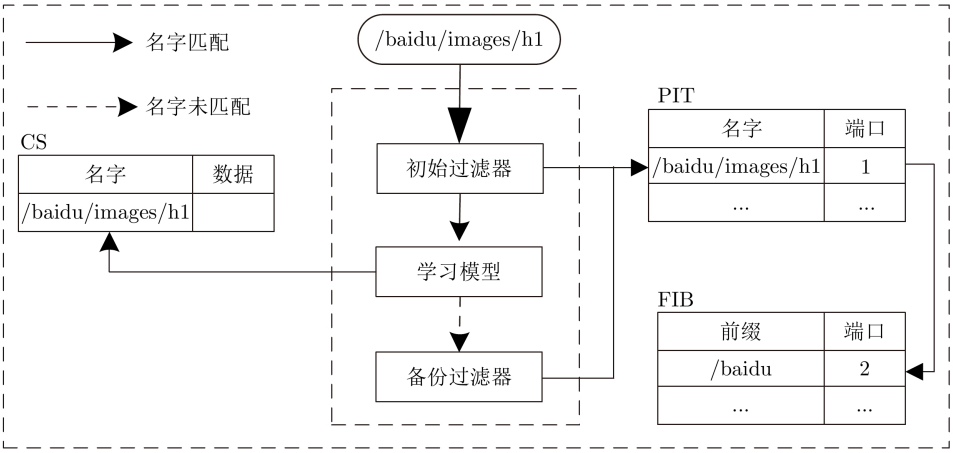

图 1 深度布隆过滤器查找结构

果在初始过滤器中判断这个名字属于内容集合中的成员或近似成员，则将名字传递给第2级进行精确查找。

第2级使用基于门控循环单元(Gated Recurrent Unit, GRU)的LSTM构建一个学习模型，将名字在学习模型中进行精确查找，由于NDN网络的层次化命名方式使得名字前缀之间有一定的语义关系，通过学习成员集合和非成员集合中的内容，可以有效预测所查找内容。因此，使用深度学习模型能够更准确地找到所请求内容的名字，但是学习模型会产生假阴性，需要一个备份过滤器消除假阴性。

第3级备份过滤器使用一个具有学习函数的布隆过滤器，利用在学习模型中设置的阈值，将在学习模型中产生的假阴性内容名字传递到备份过滤器中进行再次查找，消除假阴性，并将匹配到的内容进行转发。

在整个查找过程中，构建的学习模型能够影响查找的准确性和内存消耗，利用LSTM的门控设置解决传统循环神经网络(Recurrent Neural Network, RNN)中信息不能长期依赖的问题。构建的学习模型内部结构如图2所示。

对于每一个内容名字，在学习模型中的处理方式如下：

(1) 首先对输入的内容名字进行预处理，将名字的每个组件作为一个特征，提取出其中的一类特征；

(2) 将这些特征输入进Embedding层，把所有输入转化为向量的形式传递给LSTM层；

(3) 在LSTM层中，通过深度学习将输入内容进行解析，提取出另外一类特征；

(4) 将第1层处理的特征与通过LSTM分析提取的第2类特征进行融合，使用多层感知(MultiLayer Perceptron，MLP)模型对这两类特征进行处理；

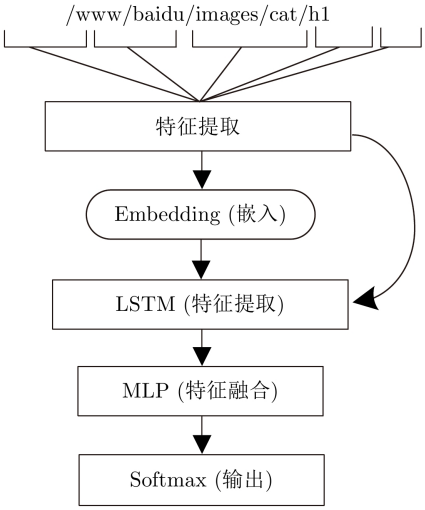

图 2 学习模型结构

(5) 将融合后的特征通过Softmax分类器进行分类处理，将处理后的内容名字输出。

通过深度布隆过滤器结构进行名字查找，对名字进行层层筛选分析，可以达到精确查找的目的，下面将通过算法详细说明整个查找过程。

2.2 深度布隆过滤器查找算法

本节描述使用深度布隆过滤器进行名字查找的算法。给定一个内容名字集合$S=\{a_1,a_2,\cdots,a_m\}$，一个非内容名字集合U，一个待查找内容名字集合Q，将初始过滤器与备份过滤器分配的总数组位数设定为bm bit，其中分配给初始过滤器b_1m bit，分配给备份过滤器b_2m bit。设置一个学习函数作为哈希函数替代传统布隆过滤器中的哈希函数，这样能够有效减少内容名字通过多个哈希函数产生的冲突，并使用机器学习中的激活函数sigmoid作为学习函数的一部分。在学习模型中，通过设置一个阈值τ对查找内容是否为内容集合中的成员做出判断。查找过程分为3级，先从第1级的初始过滤器进行查找，然后再通过基于LSTM的学习模型进行查找，最后通过备份过滤器进行最终确认，具体过程如表1所示。

表 1　面向深度布隆过滤器的名字查找算法

输入：内容集合S，非内容集合U，阈值τ
输出：内容名字x

1: 调用LSTM架构使用集合S和集合U获得一个集合D;
2: 查找x在初始过滤器中进行精确匹配;
3: **while** 对每一个$x\in S$ **do**
4:　**If** $b[i]$=1 **then**
5:　　将匹配内容x发送到深度学习模型中;
6:　**else**
7:　　将未匹配的x发送到FIB表中进行最长前缀匹配查找;
8: **end while**
9: **if** $(x,y)\in D$ **then**//在第2级深度学习模型中进行精确匹配查找
10:　计算$f(x)=\dfrac{1}{1+\mathrm{e}^{-x}}$;
11: **end if**
12: **if** $x\in S$且$f(x)<\tau$ **then**
13:　将查找获得的内容名x在路由表中对应的数据包进行转发;
14: **else if** $x\in S$且$f(x)<\tau$ **then**
15:　将未匹配的内容名字x发送到第3级备份过滤器进行查找;
16:　**while**在备份布隆过滤器查找$b[i]$=1 **do**
17:　　将查找获得的内容名x在路由表中对应的数据包进行转发;
18:　**end while**
19: **else**
20:　将第3级未匹配的内容发送到FIB表中进行最长前缀匹配查找;
21: **end if**

在名字查找过程中，根据内容集合S和非内容集合U，首先通过LSTM训练一个集合$D=\{(x_i,y_i=1)|x_i\in S\}\cup\{(x_i,y_i=0)|x_i\in S\}$，其中，$x_i$为待查找名字集合$Q$中需要查找的任一内容名字；$y_i$为通过学习模型产生的输出结果。表1的第3-5行通过初始过滤器判断查找的名字是否为集合S中的成员或近似成员，如果不是，则将内容发送给FIB，在FIB表中通过最长前缀匹配查找，确定是否有匹配的名字转发端口。如果是S集合中的成员或近似成员，将x传递给第2级的学习模型进行精确查找，如表1的第9-13行所示。作为一个预筛选，有可能会将x的相似成员传递给第2级，在第2级中设置阈值τ判定。如果$f(x)\geq\tau$，则说明x为集合S中的成员，输出x，并将这个内容名字在CS表中对应的数据包转发出去；如果$f(x)<\tau$且$x\in S$，那么说明通过学习模型查找产生了假阴性，需要将名字发送到第3级的备份过滤器进行再次查找，如表1的第14-20行所示；如果在备份过滤器中查找到该名字，则将其输出并在CS表中转发名字对应数据包，如果没有找到该名字，则在FIB表中进行最长前缀匹配查找。

在基于深度学习的3级查找结构中，利用初始过滤器进行预过滤能够大幅减少备份过滤器的内存开销。名字通过3层查找提升了查找的精度，适用于精确查找。

2.3 假阳性率分析

传统布隆过滤器通过多个哈希函数对名字进行计算，将会产生假阳性。在NDN网络中，通过CS, PIT进行精确查找，虽然能够容忍小概率的数据包转发错误，但是转发错误的内容过多就会引起网络流量的浪费，因此对内容名字的查找精度要求很高且不能占用过高的内存空间。本节将对深度布隆过滤器查找结构产生的假阳性率进行分析。

深度布隆过滤器查找结构产生的假阳性率由3个部分组成，将初始过滤器与备份过滤器分配的总数组位数设定为bm bit，其中分配给初始过滤器b_1m bit，分配给备份过滤器b_2m bit。根据传统布隆过滤器产生假阳性率的计算公式[16]可以得到，第1级初始过滤器产生的假阳性率为

$$F_o=\left(1-\left(1-\frac{1}{b_1m}\right)^{kn}\right)^k \tag{1}$$

在学习模型中，内容成员集合为S且$|S|=m$，当查找的名字不属于集合S但与成员集合中的内容非常相似时，这就产生了假阳性。因此，通过学习模型产生的假阳性率为F_p。若一个名字是成员集合中元素，但是，通过学习模型判定为非集合中元素，这就产生了假阴性。因此，名字通过学习模型产生的假阴性率为F_n。

在备份过滤器中，上一级学习模型中产生的漏报在本级会再次查找，通过学习函数进行哈希的过程仍会产生假阳性，在这个过程中产生的假阳性率为F_B。因此，可以得到整个模型所产生的总假阳性率为

$$F=F_o\left(F_p+(1-F_p)F_B\right) \tag{2}$$

根据文献[20]，对布隆过滤器产生的假阳性率进行建模，假设每个名字存储占用i bit，假阳性率以α^i下降，且当使用一个完美哈希函数时，$\alpha=1/2$。基于文献[21]，可以根据学习模型产生的假阴性数量确定备份布隆过滤器的大小。因此，只需要在备份过滤器中查找mF_n个内容名字，得到模型的假阳性率为

$$F=\alpha^{b_1}\left(F_p+(1-F_p)\alpha^{b_2/F_n}\right) \tag{3}$$

其中，b_2=b–b_1，式(3)可以转化为

$$F=\alpha^{b_1}F_p+(1-F_p)\alpha^{b/F_n}\alpha^{b_1(1-1/F_n)} \tag{4}$$

对b_1求导：

$$\begin{aligned}F(b_1)=&F_p\alpha^{b_1}\ln\alpha+(1-F_p)\left(1-\frac{1}{F_n}\right)\\&\cdot\alpha^{b/F_n}\alpha^{b_1(1-1/F_n)}\ln\alpha\end{aligned} \tag{5}$$

令$F'(b_1)=0$，可以得到最小的假阳性率，且b_2的值为

$$b_2=F_n\log_\alpha\frac{F_p}{\left(\dfrac{1}{F_n}-1\right)} \tag{6}$$

从式(6)可以看出，备份过滤器所占用的位数与分配的总位数无关，只与学习模型产生的假阳性率和假阴性率相关。因此，可以通过增大初始过滤器的数组位数降低第1级结构产生的假阳性率，从而降低第2级产生的假阳性率，这就使得备份过滤器的内存占用量大幅减少并且能够提高查找精度。

3 性能评价

本节通过实验将所提的深度布隆过滤器3级查找结构与其他几种内容名字的精确查找结构进行对比，验证所提3级查找结构在各个方面的特性。

3.1 实验设置

试验采用服务器的配置如表2所示。其中，计算机的控制核心部件CPU采用Intel Core i7处理器，CPU的核心数为6个，且拥有12个逻辑处理器，操作系统采用Windows 10。所提的深度布隆

过滤器3级查找结构代码采用Python语言实现。数据集利用Blacklist[22]数据集，选取其中64M数据作为样本集合进行训练，选取41M数据作为测试样本集合，将每个URL转变为类似“/www/baidu”形式。

从Blacklist数据集中随机选取4个分类的子文件夹，将文件中的数据进行处理筛选出其中的URL数据，然后从中选取10M数据查看其中每个内容名字的字符串长度分布情况，如图3所示。

3.2 实验结果

3.2.1 假阳性分析

本文提出的查找方法包含3级结构，第1级和第3级都使用了布隆过滤器对名字进行预处理和精确处理。为了节省内存空间并减少误报率，将两个过滤器所占用的数组位数做了分配，初始过滤器分配b_1m bit，备份过滤器分配b_2m bit。通过调整初始过滤器的系数，可以看出初始过滤器大小对查找结构假阳性率的影响如图4所示。

将两个过滤器分配的系数按照$b_1+b_2=10$的原则进行分配，通过实验可以看出，随着b_1的增大，查找结构产生的假阳性率持续降低；当给初始过滤器分配数组位超过最大限度10，给备份过滤器分配数组位为0时，假阳性率最低。这也验证了3.3节的分析，最优的假阳性率与备份过滤器的数组位数无关，备份过滤器所需的位数与学习模型产生的假阴性相关。因此，增加初始过滤器大小有利于提高查找的准确度。

表 2　服务器配置

主要模块	具体配置
主板	LENOVO-LNVNB161216
CPU	Intel Core™ i7-9750H (6核，主频2.60 GHz)
内存	DDR4 8GB (内存频率 2667 MHz)

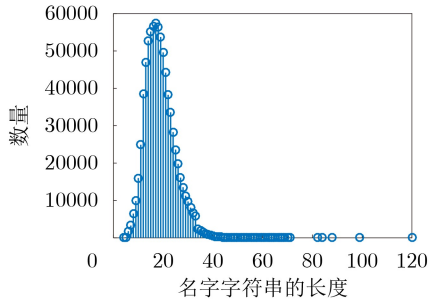

图 3 名字字符串长度分布

由于学习模型使用基于GRU的LSTM模型，因此通过实验验证GRU的大小是否会对算法的假阳性产生影响。选取了不同的学习率对不同的GRU大小进行实验，GRU和隐藏层参数配置与编号如表3所示。通过实验可以看出，当GRU的大小设置为32且隐藏层的大小设置为4时，相同学习率情况下假阴性率最小；GRU的大小设置为4时，假阴性率最高；当GRU的大小设置为8时，假阳性率最低，如图5所示。

在深度学习模型中，通过设置一个阈值τ判断内容名字的假阴性和假阳性。实验通过选取不同阈值在10M的名字表中进行判断，GRU和隐藏层参数配置与编号如表3所示。结果显示随着阈值增大，深度学习模型的假阳性率越低；反之，假阴性率就越高。当$\tau>0.5$时，通过模型查找的假阴性和假阳性率要低1到2个数量级。其中，假阴性可通过备份过滤器消除。在相同学习轮次和学习率条件下，GRU越大，假阴性和假阳性率就越小，如图6所示。

为了更直观地验证3级深度布隆过滤器查找结构名字查找的准确性，选取了1M-10M的名字表进行查找，与标准布隆过滤器和学习布隆过滤器[23]对比。深度模型选取了1个GRU和两个GRU层对比。可以看出3级深度布隆过滤器查找结构有更高的查找精度，与其他查找结构相比，假阳性率最低，如图7所示。这也验证了通过初始布隆过滤器的预过滤能够很大程度上提升内容名字查找的准确性。

3.2.2 内存消耗分析

为了验证本文结构在内存消耗方面的性能，选取了3种精确查找架构进行对比，实验结果如图8所示。其中，基于字符树的名字查找结构所占用的内存随着名字表的增大而快速增加，这是由于随着名

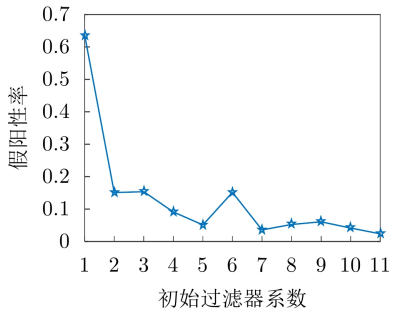

图 4 初始过滤器系数选择对假阳性率的影响

表 3　GRU和隐藏层参数配置与编号

配置编号	参数配置	配置编号	参数配置	配置编号	参数配置
I型	GRU大小=32，隐藏层大小=8	IV型	GRU大小=16，隐藏层大小=8	VII型	GRU大小=8，隐藏层大小=4
II型	GRU大小=32，隐藏层大小=4	V型	GRU大小=16，隐藏层大小=4	VIII型	GRU大小=4，隐藏层大小=4
III型	GRU大小=16，隐藏层大小=16	VI型	GRU大小=8，隐藏层大小=8	VIIII型	GRU大小=4，隐藏层大小=8

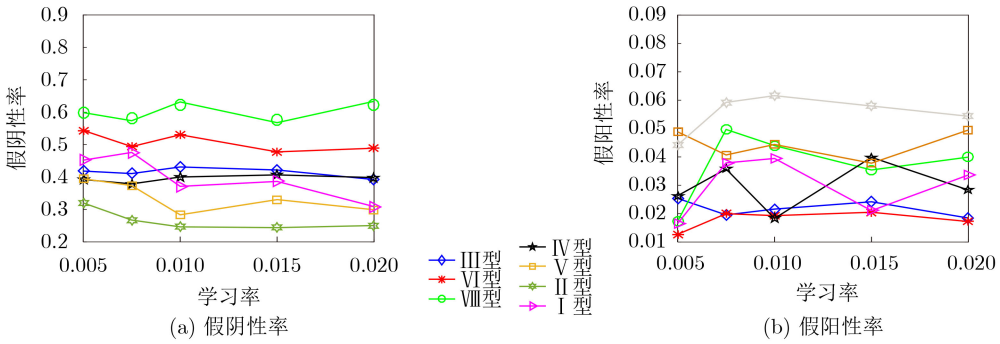

图 5 GRU大小对假阴性率和假阳性率的影响

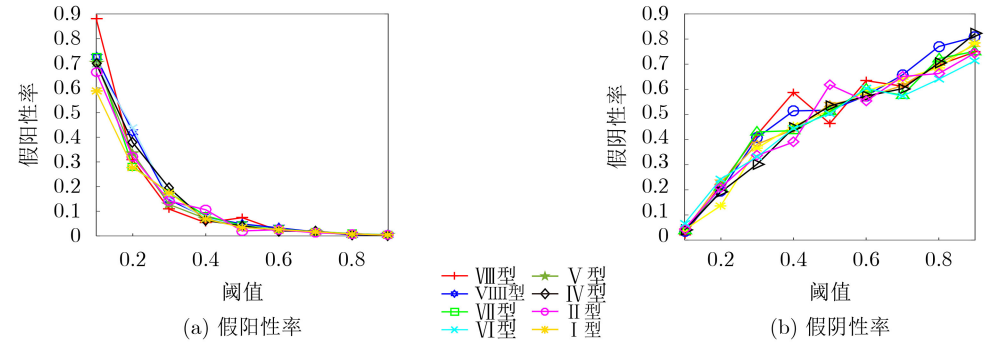

图 6 阈值τ大小对假阳性率和假阴性率的影响

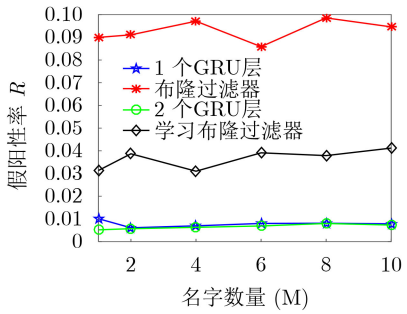

图 7 假阳性率

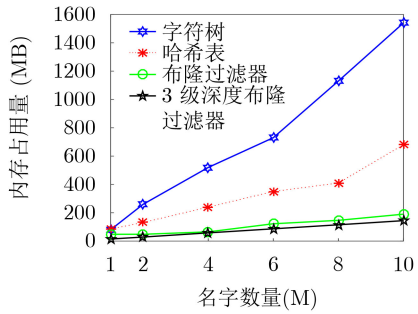

图 8 内存消耗

字数量的增多，字符树的深度不断增加，与二进制数据结构的布隆过滤器相比，内存消耗增加明显。深度布隆过滤器3级查找结构与传统布隆过滤器的内存占用接近，但优于传统布隆过滤器。与基于字符树的内容名字查找结构相比，深度布隆过滤器3级查找结构节省了近80%的内存空间，与哈希表的查找结构相比，节省了近42%的内存空间。在查找4M内容名字时，与布隆过滤器的内存占用量基本相等，但随着内容名字的增多，在名字数量达10M时，深度布隆过滤器3级查找结构的内存消耗最低。

4 结论

本文针对NDN网络中内容名字在CS和PIT表中的精确查找问题，设计了基于深度布隆过滤器的3级结构。在查找过程中，第1级使用初始过滤器对名字进行预过滤，然后通过第2级的LSTM神经网络进行精确查找和预测，最后通过第3级的备份过滤器提高查找精度。文中对3级结构产生的假阳性率进行了分析。通过实验结果可以看出，深度布隆过滤器查找结构与传统的精确查找结构相比，假阳性率大幅降低，且比传统基于字符树的查找结构节省了近80%的内存空间，因此，基于深度布隆过滤器的3级结构具有较优的查找精度和内存开销。但由于LSTM需要学习大量的样本特征，下一步计划利用小样本学习进一步提升查找效率。

参 考 文 献

[1] 杨国威, 徐泓, 李丹, 等. 未来互联网体系结构研究现状与趋势[J]. 中国基础科学, 2018, 20(3): 32–34. doi: 10.3969/

j.issn.1009-2412.2018.03.006.

YANG Guowei, XU Hong, LI Dan, *et al*. Research status and trends of future internet architecture[J]. *China Basic Science*, 2018, 20(3): 32–34. doi: 10.3969/j.issn.1009-2412.2018.03.006.

[2] 黄韬, 刘江, 霍如, 等. 未来网络体系架构研究综述[J]. 通信学报, 2014, 35(8): 184–197. doi: 10.3969/j.issn.1000-436x.2014.08.023.

HUANG Tao, LIU Jiang, HUO Ru, *et al*. Survey of research on future network architectures[J]. *Journal on Communications*, 2014, 35(8): 184–197. doi: 10.3969/j.issn.1000-436x.2014.08.023.

[3] YAO Haipeng, LI Mengnan, DU Jun, *et al*. Artificial intelligence for information-centric networks[J]. *IEEE Communications Magazine*, 2019, 57(6): 47–53. doi: 10.1109/MCOM.2019.1800734.

[4] ZHANG Lixia, AFANASYEV A, BURKE J, *et al*. Named data networking[J]. *ACM SIGCOMM Computer Communication Review*, 2014, 44(3): 66–73. doi: 10.1145/2656877.2656887.

[5] 伊鹏, 李根, 张震. 内容中心网络中能耗优化的隐式协作缓存机制[J]. 电子与信息学报, 2018, 40(4): 770–777. doi: 10.11999/JEIT170635.

YI Peng, LI Gen, and ZHANG Zhen. Energy optimized implicit collaborative caching scheme for content centric networking[J]. *Journal of Electronics & Information Technology*, 2018, 40(4): 770–777. doi: 10.11999/JEIT170635.

[6] GRITTER M and CHERITON D R. An architecture for content routing support in the Internet[C]. Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems, San Francisco, USA, 2001: 4.

[7] BARI M F, CHOWDHURY S R, AHMED R, *et al*. A survey of naming and routing in information-centric networks[J]. *IEEE Communications Magazine*, 2012, 50(12): 44–53. doi: 10.1109/MCOM.2012.6384450.

[8] 许志伟, 陈波, 张玉军. 针对层次化名字路由的聚合机制[J]. 软件学报, 2019, 30(2): 381–398. doi: 10.13328/j.cnki.jos.005572.

XU Zhiwei, CHEN Bo, and ZHANG Yujun. Hierarchical name-based route aggregation scheme[J]. *Journal of Software*, 2019, 30(2): 381–398. doi: 10.13328/j.cnki.jos.005572.

[9] FREDKIN E. Trie memory[J]. *Communications of the ACM*, 1960, 3(9): 490–499. doi: 10.1145/367390.367400.

[10] DHARMAPURIKAR S, KRISHNAMURTHY P, and TAYLOR D E. Longest prefix matching using bloom filters[J]. *IEEE/ACM Transactions on Networking*, 2006, 14(2): 397–409. doi: 10.1109/TNET.2006.872576.

[11] TAN Yun and ZHU Shuhua. Efficient name lookup scheme based on hash and character trie in named data networking[C]. Proceedings of the 12th Web Information System and Application Conference, Ji'nan, China, 2015: 130–135. doi: 10.1109/WISA.2015.72.

[12] KRASKA T, BEUTEL A, CHI E H, *et al*. The case for learned index structures[EB/OL]. https://arxiv.org/abs/1712.01208, 2020.

[13] MITZENMACHER M. Optimizing learned bloom filters by sandwiching[EB/OL]. https://arxiv.org/abs/1803.01474, 2018.

[14] LI Fu, CHEN Fuyu, WU Jianming, *et al*. Fast longest prefix name lookup for content-centric network forwarding[C]. Proceedings of the 8th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, Austin, USA, 2012: 73–74. doi: 10.1145/2396556.2396569.

[15] LI Dagang, LI Junmao, and DU Zheng. An improved trie-based name lookup scheme for named data networking[C]. Proceedings of 2016 IEEE Symposium on Computers and Communication, Messina, Italy, 2016: 1294–1296.

[16] LEE J, SHIM M, and LIM H. Name prefix matching using bloom filter pre-searching for content centric network[J]. *Journal of Network and Computer Application*, 2016, 65: 36–47. doi: 10.1016/j.jnca.2016.02.008.

[17] GOVINDARAJAN P, SOUNDARAPANDIAN R K, GANDOMI A H, *et al*. Classification of stroke disease using machine learning algorithms[J]. *Neural Computing and Applications*, 2020, 32(3): 817–828. doi: 10.1007/s00521-019-04041-y.

[18] SUTSKEVER I, VINYALS O, and LE Q V. Sequence to sequence learning with neural networks[C]. Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, Canada, 2014: 3104–3112.

[19] CHO K, VAN MËRRIENBOER B, GULCEHRE C, *et al*. Learning phrase representations using RNN encoder-decoder for statistical machine translation[C]. Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 2014: 1724–1734. doi: 10.3115/v1/D14-1179.

[20] BLOOM B H. Space/time trade-offs in hash coding with allowable errors[J]. *Communications of the ACM*, 1970, 13(7): 422–426. doi: 10.1145/362686.362692.

[21] BRODER A and MITZENMACHER M. Network applications of bloom filters: A survey[J]. *Internet Mathematics*, 2004, 1(4): 485–509. doi: 10.1080/15427951.2004.10129096.

[22] Blacklist[DB/OL]. http://squidguard.mesd.k12.or.us/blacklists.tgz.2020.7.5.

[23] WANG Qianyu, WU Qingtao, ZHANG Mingchuan, *et al*. Learned bloom-filter for an efficient name lookup in information-centric networking[C]. Proceedings of 2019 IEEE Wireless Communications and Networking Conference, Marrakesh, Morocco, 2019: 1–6.

吴庆涛：男，1975年生，教授，研究方向为云计算、物联网、下一代网络.

师君如：女，1997年生，硕士生，研究方向为信息中心网络.

张明川：男，1977年生，教授，研究方向为物联网、下一代网络、机器学习.

王倩玉：女，1991年生，硕士生，研究方向为信息中心网络.

朱军龙：男，1982年生，副教授，研究方向为人工智能、机器学习、新型网络.

张宏科：男，1957年生，教授，研究方向为下一代网络、智慧协同网络.

责任编辑：陈 倩