

文章编号:1007-5321(2020)05-0027-07

DOI:10.13190/j.jbupt.2020-084

面向多模态数据的混合型 FIB

王彬志¹, 李卓¹, 罗蓬², 马天祥², 刘开华¹

(1. 天津大学 微电子学院, 天津 300072; 2. 国网河北省电力有限公司 电力科学研究院, 石家庄 050021)

摘要: 针对未来网络转发信息库(FIB)中多模态数据带来的差异化快速索引、高效存储转发信息和多模态数据最长前缀匹配等问题,设计了一种支持多模态数据索引的混合型 FIB,称为 Hybrid-FIB. 通过对不同类型的数据进行差异化处理,得到可供神经网络模型学习的输入向量,进而训练出能够实现均匀分布的神经网络混合索引模型. 为了实现多模态数据最长前缀匹配,在片上静态随机存取存储器中部署两组 Hybrid-FIB 结构. 实验结果表明,该混合型 FIB 在误判率、存储消耗及吞吐量等方面具备优异性能.

关键词: 信息中心网络; 多模态网络; 转发信息库; 神经网络

中图分类号: TP393.0

文献标志码: A

A Hybrid Forwarding Information Base for Multi-Modal Data

WANG Bin-zhi¹, LI Zhuo¹, LUO Peng², MA Tian-xiang², LIU Kai-hua¹

(1. School of Microelectronics, Tianjin University, Tianjin 300072, China;

2. Electric Power Research Institute, Hebei Electric Power Corporation, Shijiazhuang 050021, China)

Abstract: In order to solve the problems of rapid indexing, efficient storage of forwarding information and longest prefix matching brought by multi-modal data in the forwarding information base(FIB) in the future network, a hybrid FIB based on neural networks, called Hybrid-FIB, which supports multi-modal data indexing is designed. Hybrid-FIB differentiates different type of data to obtain input vectors for neural network model, and then trains a neural network hybrid index model that can achieve uniform distribution. Experiments show that deploying two sets of Hybrid-FIB on the static random access memory can not only achieve the longest prefix matching of the multi-modal data, but also have better retrieval speed and misjudgment rate than the current network.

Key words: information-centric networking; polymorphic network; forwarding information base; neural network

随着网络技术的不断演进,全息通信、车联网、虚拟现实等创新型的网络应用不断涌现. 然而当前的网络架构已经不能满足用户对互联网内容化、个性化、泛在移动、安全隐私等方面的需求. 为此,近年来国内外学术界和产业界不断深化未来网络架构

的研究,以内容标识、空间坐标标识、身份标识等为中心的 novel 寻址路由技术得以迅猛发展,并在部分领域获得认可. 邬江兴院士团队于 2019 年提出全维可定义的多模态智慧网络体系^[1]. 其中,多模态寻址路由是多模态智慧网络的关键要素,体现为基

收稿日期: 2020-07-10

基金项目: 国家自然科学基金项目(61602346); 河北省重点研发计划项目(20314301D); 天津大学北洋学者青年骨干教师项目(2020XRG-0102)

作者简介: 王彬志(1997—),男,硕士生.

通信作者: 李卓(1984—),男,副教授,硕士生导师, E-mail: zli@tju.edu.cn.

于内容、身份、地理空间等标识的多种寻址路由模式。华为研究团队于 2019 年发布了针对网络 5.0 架构 NewIP 方案^[2]。该方案通过实现变长网络地址、多样化寻址、面向服务的路由等,支持海量异构通信主体、异构网络间的互联互通。思科团队于 2017 年提出混合信息中心网络(hICN, hybrid information-centric networking)组网方案^[3],不使用任何封装或隧道技术,将信息中心网络(ICN, information-centric networking)功能引入现有网际互联网协议(IP, internet protocol),并在网络中进行增量部署。张宏科教授团队提出了智慧协同标识网络体系^[4],提出资源动态适配的“三层”“两域”体系结构模型。上述方案的核心都是通过多模态网络路由标识,研究高效、安全、可演进组网转发技术,建立智慧化、灵活化、内生安全的网络架构。

当前,在推动互联网增量式部署和演进式发展的同时,充分吸收和利用新思路、新方法,对互联网技术体系进行基础性变革,构建面向 2035 年的新型网络体系架构和技术体系将成为大势所趋^[5]。鉴于当前多模态网络的组成架构及寻址方式尚未明确,对于多模态数据差异化处理或是归一化处理也尚未形成统一论,对此类问题不做过多讨论,而是自底层向上看,试图通过使用机器学习技术为将来可能出现的名称数据和 IP 地址共存的转发信息库(FIB, forwarding information base)提供一种混合数据处理能力。IP 地址和内容名称是多模态网络中地址、内容、身份、地理空间等寻址标识的 2 种典型代表,对其进行研究是对于多模态网络中关键技术研究的有益尝试。为此,提出了 Hybrid-FIB 存储结构,能够有效支持多模态数据检索,并且该存储结构性能优异,能够满足路由器各项性能的要求。

1 Hybrid-FIB 的设计需求

在 IP 网络中,一条 FIB 条目一般包括目的地址、前缀长度、传输端口、下一跳地址、表明路由特征的标志以及时间戳。另外,为了缓解 IP 地址空间耗尽问题,当前的无类域间路由编制方案摒弃了传统的基于类的地址分配方式,使用任意长度的地址前缀。尽管这个方案十分有效地提高了地址空间的利用率,但是使用任意长度的地址前缀给实现最长前缀匹配(LPM, longest prefix matching)带来了极大的挑战。与 IP 网络中 FIB 的表项不同,ICN 网络中 FIB 条目的主要形式为:〈name prefix, stale time, in-

terface ID, routing preference, RTT, status, rate limit〉。其中,条目中的名称前缀为变长、无边界的名称字符串,其形式类似于 IP 网络下的统一资源定位器(URL, uniform resource locator),以分隔符为断点,形成不同粒度的名称前缀。实现变长名称字符串的最长名称前缀匹配(LNPM, longest name prefix matching)是 ICN 路由器 FIB 设计的一个难点。另外,不同应用驱动下的名称数据也极具个性化,并且未来路由器转发表的数据量也达到了百万级别。如何实现压缩存储,将 FIB 部署在具有较快处理速度的静态随机存取存储器(SRAM, static random access memory)上是 ICN 路由器 FIB 设计的又一难点。

为了实现多模态数据的混合索引,Hybrid-FIB 不仅要满足上述的需求,还应当满足混合表项下的各种需求。首先,在混合表项中,固定长度的 IP 地址与变长的名称数据混合,使表项的构成更加复杂。比起数据类型单一的网络,多模态网络下 FIB 表的快速、准确检索成为 Hybrid-FIB 难点之一;其次,由于多模态网络中 2 种数据的混合会大大增加 FIB 的存储消耗,而将 Hybrid-FIB 部署在具有较大存储空间动态随机存取存储器(DRAM, dynamic random access memory)上不能满足未来用户对于快速检索的需要,因此,如何设计 Hybrid-FIB 存储结构,最大程度地压缩混合表项数据,并能够部署在具有较快处理速度的 SRAM 上成为 Hybrid-FIB 设计的难点之二;最后,由于 IP 网络中需要任意 IP 地址前缀长度实现最长前缀匹配,ICN 中需要不同粒度的名称前缀实现最长前缀匹配,这 2 种最长前缀匹配方式的不同,如何实现不同数据下最长前缀匹配成为 Hybrid-FIB 设计的难点之三。

2 Hybrid-FIB 原理及结构

2.1 Hybrid-FIB 基本原理

面对未来多模态网络百万级别数据量的转发平面,FIB 存储结构的误判率是一个重要考察因素。哈希函数编码的随机性导致哈希映射无规律,存在较大误判率。使用 Learning-Tree 索引单元^[6]的 FIB 存储结构通过反向传播神经网络(BPNN, back propagation neural networking)学习索引内容在存储器中的累积分布函数(CDF, cumulative distribution function),利用 CDF 的均匀分布特性实现映射表的均匀映射,其映射效果等同于一个完美的哈希函数。Learning-Tree 的研究成果表明,神经网络索引模型

的输入为 5 维向量时,索引单元的误利率、存储消耗和索引速度的综合效果达到了最优^[6].

Hybrid-FIB 使用 Learning-Tree 的基本思想,通过对不同模态的数据进行差异化处理,生成可供 Learning-Tree 学习的输入向量. 通过合理地构建神经网络训练集,训练出能够实现多模态数据索引的混合索引模型. 该混合索引模型的映射更均匀,极大地降低了映射冲突,提高了存储效率,且该索引的结构可以实现同类相近数据的聚合,满足未来对于路由聚合的潜在需求.

混合索引模型包括 3 个基本单元,分别为数据处理单元、数据索引单元和映射单元,如图 1 所示. 数据处理单元用于处理输入的多模态数据,将定长的 IP 地址和变长的名称数据处理为 5 维输入向量. 对于内容名称,假设一个变长的内容名称共包含 n 个字符,可以将该名称视为一个 n 维向量 \mathbf{x} . 如果名称长度 $n \leq 5$,则输入向量 \mathbf{y} 的任意一维 $y_i = x_i (0 \leq i < n), y_i = 0 (n \leq i <$

5); 如果名称长度 $n > 5$,将名称向量 \mathbf{x} 拆分为多个 5 维子向量 $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}, \dots, \mathbf{x}^{(m)}, y_i = \text{Bitxor}(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, \mathbf{x}_i^{(3)}, \dots, \mathbf{x}_i^{(m)}) (0 \leq i < 5)$,最终得到 5 维输入向量 \mathbf{y} . IP 地址在用点分十进制表示时,可以将其视为 4 维向量,神经网络对于频繁变化的地址后两位不敏感,因此使用逆序排列以提高神经网络对于 IP 地址的索引准确率,并在其首维添加 0,得到 5 维输入向量. 数据索引单元为经过训练的小型 BP 神经网络搭建的两层塔式结构,包括第 1 层 1 个 BPNN,第 2 层 1 000 个 BPNN,用来学习索引内容在存储器中的分布情况. 其中, $\text{BPNN}_{i,j}$ 表示第 i 层第 j 个 BPNN,第 1 层 BPNN 将转发平面百万数据分为 1 000 类,第 2 层 BPNN 学习数据在存储器中的分布情况. 映射单元部署一个 Bitmap 结构,将索引单元学习出的 CDF 函数值映射到实际存储空间,进而得到转发数据在实际存储器中的存储位置.

2.2 Hybrid-FIB 存储结构

鉴于网络转发平面对于快速转发的需求以及当

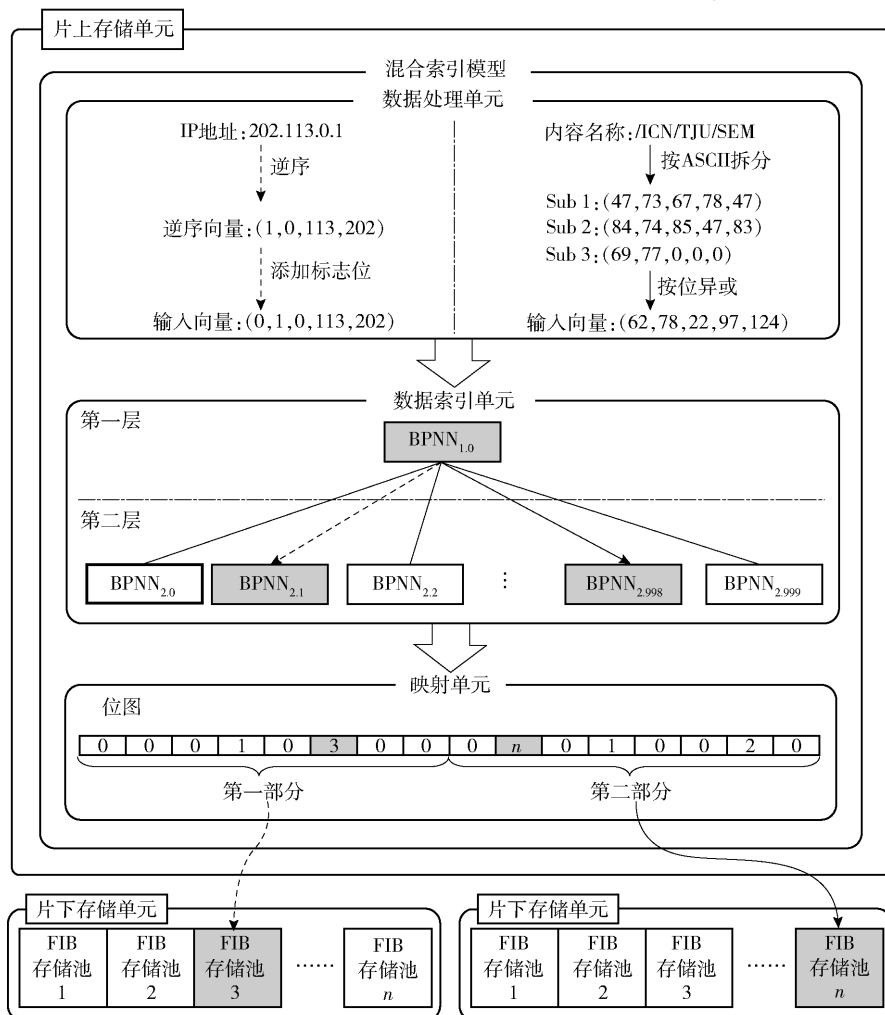


图 1 Hybrid-FIB 整体结构

前快速存储器存储水平,Hybrid-FIB 存储结构设计为片上和片下 2 个存储单元,具体结构如图 1 所示.片上存储单元使用 SRAM 以保证转发平面对于快速索引的需求,并在其上部署了一系列对应不同模式数据长度的混合索引模型,以实现基于 IP 地址的 LPM 机制和基于名称数据的 LNPM 机制.片下存储单元使用 DRAM 以应对转发平面百万级别数据量的存储消耗,并在其上部署一系列与混合索引结构对应的 FIB 存储池用来存储实际的转发信息.当多模式数据索引到同一地址发生误判时,将采用链地址法在片外存储器内容存储池中处理冲突.

经过统计,IP 地址前缀与名称数据前缀长度占比不均匀,因此应当按照前缀长度的比例合理地分配存储空间.图 2 所示为 IP 地址前缀和名称数据前缀的统计结果,其中 IP 地址中前缀长度为 24 bit 的最多,小于 8bit 的前缀占比几乎为 0,长度为 0 ~ 21 的前缀数量占 19%,长度为 22 ~ 32 的前缀数量占 80.97%.名称数据中组件个数为 4 的最多,占到了 82.34%,其余组件个数的名称共占了 17.66%.考虑到片上存储器的有限存储空间,并且为了实现存储空间的合理分配,Hybrid-FIB 的片上存储单元部署两组混合索引结构,其中一组对应 IP 前缀长度为 0 ~ 21 的 IP 地址和名称组件个数为 4 的名称数据;另一组则对应 IP 地址长度为 22 ~ 32 的 IP 地址和名称组件个数为除 4 个以外其他个数的名称数据.

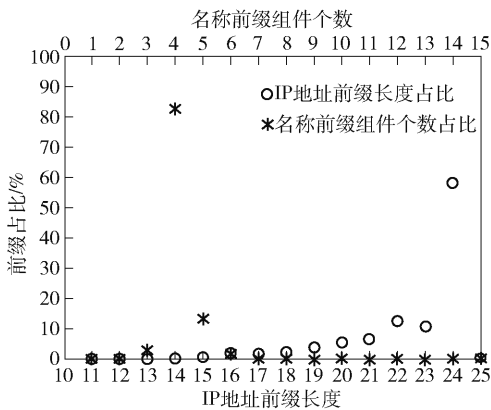


图 2 IP 地址前缀与名称数据前缀分布

3 Hybrid-FIB 检索算法及检索实例

3.1 Hybrid-FIB 检索算法

无论是 IP 数据包还是内容数据包到达路由器时,Hybrid-FIB 都会对其执行相应的检索算法.其

检索算法流程如算法 1 所示.由于 IP 地址和名称数据进行最长前缀匹配的方式不同,所以当路由器收到一个数据包时,需要先解析包头,判断输入的数据包是 IP 数据包还是内容数据包,进而进行相应的匹配及转发操作.

如果收到的是 IP 数据包,则将 IP 前缀以 21 位前缀为分界,分为前缀长度为 1 ~ 21 位,22 ~ 32 位两部分,按照前缀长度由高到低分别输入两组 Hybrid-FIB 中,并行执行前缀匹配操作.如果存在匹配结果,则最长的地址前缀为最长匹配前缀,进而读取该匹配项对应的转发信息进行数据包转发.如果不存在匹配项,则输出检索结果“不存在匹配信息”.如果收到的数据包为内容数据包,则将解析出的名称按照名称组件个数分为 4 个组件和其他个数组件两部分,并分别输入两组 Hybrid-FIB 中,并行执行前缀匹配操作.如果存在匹配结果,则组件个数最多的为最长名称匹配前缀,进而读取该匹配项对应的转发信息进行数据包转发.如果不存在匹配项,则输出检索结果“不存在匹配信息”,这表明路由器不存在该数据包的转发信息.

算法 1 多模式数据包转发信息检索过程

```

1 Procedure Lookup( Multi-modal Data  $x$  )
2   if  $x$  == Name
3     Input ← Bitxor(  $x$  );
4   else
5     Input ← Reverse(  $x$  );
6   end if
7   L1_Output ← BPNN1.0( Input );
8   L2_Output ← BPNN2.L1_Output( Input );
9   Pos ← Floor( L2_Output × Bitmap.size );
10  if Bitmap[ Pos ] ≠ 0 then /* 读取转发信息 */
11    Part ← Pos / Part_Size;
12    Offset_Addr ← Bitmap[ Pos ];
13    Memory_Addr ← Base_Addr[ Part ] + Offset_Addr;
14    return FIB_Store[ Memory_Addr ];
15  else /* 输出无法匹配 */
16    Failed();
17  end if
18 End Procedure

```

Hybrid-FIB 可以根据路由表进行路由数据信息

更新,更新主要包括插入、修改和删除 3 种操作,更新过程如算法 2 所示。

算法 2 多模态数据包转发信息更新过程

```

1  Procedure Update ( Multi-modal Data  $x$ , Operation)
2  if  $x = \text{Name}$ 
3    Input  $\leftarrow \text{Bitxor}(x)$ ;
4  else
5    Input  $\leftarrow \text{Reverse}(x)$ ;
6  end if
7  L1_Output  $\leftarrow \text{BPNN}_{1,0}(\text{Input})$ ;
8  L2_Output  $\leftarrow \text{BPNN}_{2, \text{L1\_Output}}(\text{Input})$ ;
9  Pos  $\leftarrow \text{Floor}(\text{L2\_Output} \times \text{Bitmap.size})$ ;
10 Part  $\leftarrow \text{Pos} / \text{Part\_Size}$ ;
11 switch Operation do
12   case INSERT /* 插入 */
13     Bitmap[Pos].insert();
14     Memory_Addr  $\leftarrow \text{Base\_Addr}[\text{Part}] + \text{Bitmap}[\text{Pos}]$ ;
15     FIB_Store[Memory_Addr].insert();
16   case MODIFY /* 修改 */
17     Memory_Addr  $\leftarrow \text{Base\_Addr}[\text{Part}] + \text{Bitmap}[\text{Pos}]$ ;
18     FIB_Store[Memory_Addr].modify();
19   case DELETE /* 删除 */
20     Memory_Addr  $\leftarrow \text{Base\_Addr}[\text{Part}] + \text{Bitmap}[\text{Pos}]$ ;
21     Bitmap[Pos] = 0;
22     FIB_Store[Memory_Addr].delete();
23   endsw
24   endsw
25 End Procedure

```

当 FIB 表中的转发信息需要更新时,需要判断数据包的类型,若是内容数据包,则进行按位异或处理;若是 IP 地址,则对 IP 地址进行逆序处理并添加标志位。当执行插入操作时,需要进行最长前缀匹配,将该数据插入对应长度的混合索引模型中,然后根据索引模型的输出结果计算偏移地址,访问片下的 FIB 存储池,将对应的路由转发信息存储到该地址;当执行修改操作时,对于输入数据执行精确名称匹配(ENM, exact name matching),根据索引模型的输出结果访问片下 FIB 存储池,并对路由信息执行修改操作;当执行删除操作时,同样执行精确名称匹

配,访问片下 FIB 存储池,并同时删除混合索引模型和 FIB 存储池中的对应信息。

3.2 数据检索实例

图 1 所示的虚线和实线箭头指向的流程示出了多模态数据在 Hybrid-FIB 中的索引实例。对于一个到达的多模态数据包,首先需要判断数据包的类型,如果到达的是 IP 数据包(虚线箭头指示),则混合索引模型数据处理单元对 IP 地址进行逆序操作,并在首位添加标志位,得到输入向量(0, 1, 0, 113, 202)。将该输入向量输入到数据索引单元,经过第 1 级神经网络计算得到分类编号为 1,则由第 2 级神经网络 BPNN_{2,1}计算 CDF 值。假设计算得到的 CDF 值为 0.4,乘以槽的个数 15 即为该 IP 数据包的映射位置 $0.4 \times 15 = 6$,该位置位于 Bitmap 的第 1 部分,其记录的数字序号为 3,因此,实际的地址为第 1 部分对应的基地址加地址偏移量 3,通过访问片下存储器,找到对应的 FIB 存储池 3 读取转发信息。如果到达的是名称为/ICN/TJU/SEM 的内容数据包(实线箭头所指),混合索引模型的数据处理单元需要对名称进行拆分并执行 5 维异或运算,得到输入向量(62, 78, 22, 97, 124)。将该输入向量输入第 1 级神经网络,得到分类值为 998,则在第 2 级神经网络 BPNN_{2,998}计算 CDF 值。假设结果为 0.667,乘以槽的个数 15 即为该名称的映射位置 $0.667 \times 15 \approx 10$,该位置位于 Bitmap 的第 2 部分,其记录的数字序号为 n ,因此,实际的地址为第 2 部分对应的基地址加地址偏移量 n ,对应访问到片下 FIB 存储池 n 读取转发信息。

4 Hybrid-FIB 性能评价

4.1 实验设置

误判概率、存储消耗和吞吐量是 FIB 性能评价的 3 个重要方面,因此设计了实验与基于二进制树的 Binary Patricia Trie-FIB、基于哈希表 Hash-FIB 和基于 B-MBF 的 B-MaFIB 进行了性能对比,其中 Hash-FIB 中的哈希函数分别选用 MD5、SHA1、City-Hash256 进行测试。为了模拟真实场景下的索引情况,实验中的测试集使用了原始未经处理的数据,并采用数据量为 50、100、150、200 万的随机多模态数据测试效果。

所有实验和测试全部在一台配置为 Intel Xeon E5-1650 v2 3.50 GHz、DDR3 24 GB SDRAM 的小型工作站上完成。神经网络索引模型的构建使用 Mat-

lab 神经网络工具箱实现. 训练完成后提取其中的参数,使用 C++ 语言实现 Hybrid-FIB 设计和性能评估. 同样,对比实验中 Binary Patricia Trie-FIB、Hash-FIB 和 B-MaFIB 同样使用 C++ 语言实现.

Hybrid-FIB 中,每个 BP 神经网络由 5 个输入神经元、20 个隐藏层神经元和 1 个输出神经元组成;Bitmap 每个槽的大小设置为 2 byte. Binary Patricia Trie-FIB 中,每个地址指针为 4 byte. Hash-FIB 中的哈希函数每个指针大小为 4 byte. B-MaFIB 中, Bloom filter 的大小设置为 2^{24} bit,定位数组的大小设置为 24 bit,Bitmap 每个槽的大小同样为 2 byte.

4.2 误判率

图 3 所示为 Hybrid-FIB 对多模态数据索引的误判率. 由于 Binary Patricia Trie-FIB 方案不存在误判率,所以在比较误判率时未列出该方案.

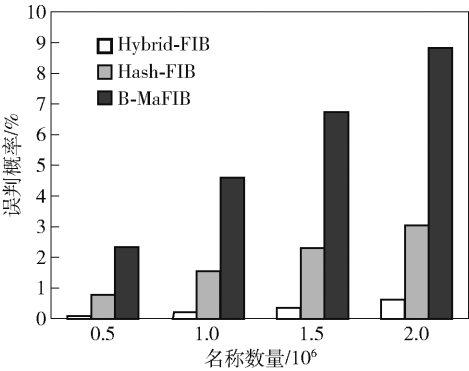


图 3 Hybrid-FIB 误判率

实验中将槽个数控制为 3 200 万个,依次向 Hybrid-FIB、Hash-FIB、B-MaFIB 中插入 50、100、150、200 万个多模态数据,Hybrid-FIB 的误判率分别为 0.049%、0.186%、0.362% 和 0.580%,其他方案的结果如表 1 所示. 即使面对 200 万的转发平面数据量,Hybrid-FIB 的误判率也远低于当前互联网丢包率小于 1%^[7]的需求,而此时 Hash-FIB 及 B-MaFIB 的误判率却已远大于 Hybrid-FIB.

表 1 Hybrid-FIB 与其他方案的误判率对比

名称数量/ 10^6 个	Hybrid-FIB/%	Hash-FIB/%	B-MaFIB/%
0.5	0.049	0.782	2.350
1.0	0.186	1.561	4.637
1.5	0.362	2.314	6.789
2.0	0.580	3.070	8.906

4.3 存储消耗

由于数据的路由转发信息存储在片下 FIB 存储

池中,这些转发信息消耗的存储空间是相同的,并且该存储空间的消耗不会对 Hybrid-FIB 能否部署在片上造成影响. 因此主要对比分析各种索引结构的存储消耗.

Hybrid-FIB、Hash-FIB、B-MaFIB 索引结构的存储消耗主要是由映射表的大小决定的. 为了满足网络 1% 丢包率的需求,Hash-FIB、B-MaFIB 需要更多的槽个数来降低冲突率. 3 种结构的内存消耗如表 2 所示. 当多模态数据的数量达到 200 万时,Hybrid-FIB 需要 3 100 万个槽,而 Hash-FIB 与 B-MaFIB 所需要的槽个数比 Hybrid-FIB 大 1~2 个数量级. 这是由于神经网络学习出了数据的分布规律,并实现了数据的均匀映射,比其他数据结构,Hybrid-FIB 的冲突率很小,仅需要较少的槽个数就能够把误判率降低到 1%.

表 2 不同数据量下映射表所需槽的总数 10^6 个

名称数量	Hybrid-FIB		Hash-FIB	MaFIB
	Bitmap(1)	Bitmap(2)		
0.5	1	3	25	144
1.0	3	8	51	576
1.5	6	15	75	640
2.0	8	23	101	1 152

对于 50、100、150、200 万个多模态数据量,Bitmap 的存储消耗分别为 8、22、42 和 62 MB. 数据索引结构中每个 BP 神经网络模型的存储消耗为 $(20 \times 5 + 20 \times 1 + 1 \times 20 + 1 \times 1) \times 8 \times 2\,002 \approx 2.258$ MB. 因此,对于上述不同的数据量,混合索引结构索引模型的存储消耗总计分别为 10.258、24.258、44.258 和 64.258. 表 3 所示为 Hybrid-FIB 与其他索引结构存储消耗的情况,其中 Hybrid-FIB 和 Binary Patricia Trie-FIB 方案的存储消耗远远低于其他方案. 这是因为 Hybrid-FIB 通过学习出存储器的存储规律实现了均匀映射,从而减小了存储消耗,而 Binary Patricia Trie-FIB 通过集成相同前缀的方式降低了存储消耗.

表 3 Hybrid-FIB 与其他方案的存储消耗对比

名称数量/ 10^6 个	Hybrid-FIB/ MB	Binary Patricia Trie-FIB/MB	Hash-FIB/ MB	B-MaFIB/ MB
0.5	10.258	16.265	100	292.097
1.0	24.258	32.507	204	1 156.097
1.5	44.258	48.762	300	1 284.097
2.0	64.258	65.046	404	2 308.097

当前,一个 4 SRAM 线卡的存储空间最大为 128.746 MB,因此 Hybrid-FIB 存储结构能够部署在 SRAM 上,而 Hash-FIB 和 B-MaFIB 存储结构都需要较大的存储空间,难以部署在 SRAM 上。

4.4 吞吐量

Hybrid-FIB 与其他方案吞吐量的对比如图 4 所示,可见 Hybrid-FIB 和 Binary Patricia Trie-FIB 的吞吐量远大于其他 2 个方案,Hybrid-FIB 的吞吐量更是达到了 11.68 MP/s。这是因为尽管 Hybrid-FIB 中索引单元神经网络的训练过程极其复杂,但是实际索引时只需考虑测试过程。将训练好的神经网络参数提取出来并进行运算,其映射效果趋于一个完美的哈希函数,运算速度与 CityHash 相当,远高于 MD5、SH1 哈希方法。另外,索引模型选用的 BP 神经网络结构简单,运行速度快,且能够实现均匀映射,极大地压缩了存储空间,因此,该索引模型能够部署在片上高速存储器。由于片下存储器仅用来存储转发信息,实际检索速度完全取决于片上索引模型的检索速度。相较 Hash-FIB,由于哈希函数随机映射使映射不均匀,存储空间消耗较大,难以部署在片上快速存储器,所以速度较低。相较 Binary Patricia Trie-FIB,尽管 Binary Patricia Trie-FIB 也能够部署在片上存储器,但是其较多的层数影响了检索速度。

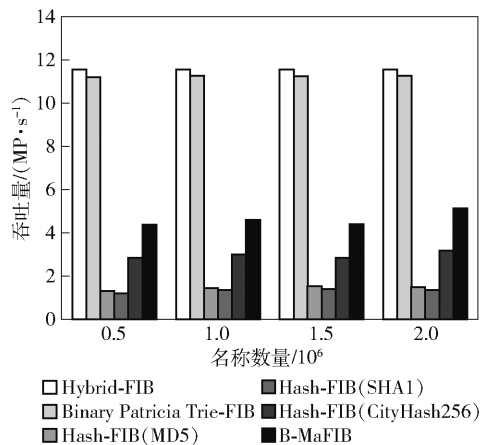


图4 Hybrid-FIB 吞吐量

5 结束语

提出了一种支持多模态数据索引的 Hybrid-FIB 存储结构。通过使用 IP 地址与名称数据构成的多模态数据训练神经网络模型,得到了混合索引模型。该模型通过对 IP 地址和名称数据进行特定处理,学习

出服从均匀映射的分布规律 CDF 函数,极大地提高了存储效率。通过在片上部署 2 个混合索引结构实现多模态数据检索,不仅在保证检索速度的基础上最大程度地减小了存储消耗,而且有效地支持了 IP 地址及内容名称多模态数据的最长前缀匹配机制。实验结果表明,在保证网络小于 1% 丢包率的前提下,对于百万级别的数据量,Hybrid-FIB 索引结构的存储消耗为 64.258 MB,吞吐量为 11.68 MP/s。在未来工作中,将在实际通信环境下测试 Hybrid-FIB 的各项性能指标,并进一步研究 Hybrid-FIB 的部署方案。

参考文献:

- [1] 胡宇翔, 伊鹏, 邬江兴, 等. 全维可定义的多模态智慧网络体系研究[J]. 通信学报, 2019, 40(9): 1-12.
Hu Yuxiang, Yi Peng, Wu Jiangxing, et al. Research on the full-dimensional defined polymorphic smart network [J]. Journal on Communications, 2019, 40(9): 1-12.
- [2] 郑秀丽, 蒋胜, 王闯. NewIP: 开拓未来数据网络的新连接和新能力[J]. 电信科学, 2019, 35(9): 2-11.
Zheng Xiuli, Jiang Sheng, Wang Chuang. NewIP: new connectivity and capabilities of upgrading future data network [J]. Telecommunications Science, 2019, 35(9): 2-11.
- [3] Carofiglio G, Muscariello L, Auge J, et al. Enabling ICN in the internet protocol: analysis and e-evaluation of the Hybrid-ICN architecture [C] // Proceedings of the 6th ACM Conference on Information-Centric Networking. Macao: ACM, 2019: 55-66.
- [4] Zhang Hongke, Dong Ping, Yu Shui, et al. A scalable and smart hierarchical wireless communication architecture based on network/user separation [J]. IEEE Wireless Communications, 2017, 24(1): 18-24.
- [5] 李军飞, 胡宇翔, 邬江兴. 面向 2035 的多模态智慧网络技术发展路线图[J]. 中国工程科学, 2020, 22(3): 141-147.
Li Junfei, Hu Yuxiang, Wu Jiangxing. Development roadmap of polymorphic intelligence network technology toward 2035 [J]. Strategic Study of CAE, 2020, 22(3): 141-147.
- [6] Yan Liu, Li Zhuo, Liu Kaihua. Learning tree: neural network-based index for NDN forwarding plane [C] // Proceedings of the ACM SIGCOMM 2019 Conference Posters and Demos. Beijing: ACM, 2019: 63-65.
- [7] You Wei, Mathieu B, Truong P, et al. DIPIT: a distributed bloom-filter based PIT table for CCN nodes [C] // 2012 21st International Conference on Computer Communications and Networks (ICCCN). Munich: IEEE, 2012: 1-7.