

一种基于 Trie 的快速 IP 路由查找算法

郜国良, 李广军

(电子科技大学 通信与信息工程学院, 四川 成都 611731)

摘 要: Internet 的飞速发展要求核心路由器能够实现快速的分组转发和路由更新功能, 实现这一功能的关键是路由表的组织结构和快速的路由查找算法. 提出了带有转发域信息树的多分支 Trie 结构路由查找算法, 它由固定步长的多分支 Trie 结构的路由表和转发域信息树两部分组成. 对于一个长度为 w 的路由前缀, 其查找、插入、删除路由的时间复杂度均为 $O((w-m)/n+1)$, 其中 m, n 为 Trie 树的步长. 它解决路由查找过程中快速更新的问题, 具有算法简单、查找速度快、易于更新、空间利用率高、便于向 IPv6 过渡等优点.

关键词: 最长前缀匹配; Trie 树; 快速查找; 快速更新

中图分类号: TP301.6

文献标识码: A

文章编号: 1000-7180(2011)06-0163-05

A Fast Routing Lookup Algorithm Based Trie

GAO Guo-liang, LI Guang-jun

(School of Communication and Information Engineering,
University of Electronic Science and Technology of China, Chengdu 611731, China)

Abstract: With rapid development of Internet, it orders the core router to have the ability of high speed packets forwarding and fast updating. The key to implement this function is data structure of router table and lookup algorithm. This paper proposes a new scheme based multibit Trie tree with a next-hop tree. It consists of a fixed stride routing table based Trie tree and a next-hop tree. For a routing prefix whose length is w , the time for search, insertion, deletion operation are all $O((w-m)/n+1)$, where m and n are both stride of Trie tree. It solves IP lookup problem with fast updates and has the advantage of simple algorithm, fast lookup and updating easy, and high memory utilization and transition to IPv6 convenient.

Key words: longest matching prefix; Trie tree; fast lookup; fast update

1 引言

路由器的主要功能是按照 IP 分组中的目的地址转发分组. 目前路由器的主要性能瓶颈是在转发表中查找某个路由所消耗的时间, 因此快速的 IP 地址查找算法是实现高速分组转发的关键. 近年来, 随着 Internet 的普及和多媒体业务的涌现, 网络流量在主要交换节点每隔几个月就会出现成倍的增长. 为了满足网络流量快速增长的要求并提供良好的 QoS^[1-2] 服务, 迫切需要一种具有快速更新能力的快速 IP 路由查找算法. 另外, Internet 是一个动态的

系统, 随时都会有新的路由器和子网加入到网络中, 在核心路由器中, 路由表的更新比较快, 通常更新在每秒几百条, 甚至上千条^[3] 以上. 因此, 路由查找算法除完成快速查找外, 还必须能够实现快速的更新功能. 在设计 IP 路由查找算法时除了满足查找速度、更新时间的要求外, 还必须考虑存储器容量大小的因素, 实施方案所需的存储器应尽可能的小, 在一定的空间中存储尽可能多的路由前缀, 以便于实现和满足不断增长的前缀数目的要求.

事实上, 对 IP 路由查找算法来讲, 很难同时满足查找速度、更新速度、空间利用率等三方面的要

收稿日期: 2010-07-28; 修回日期: 2010-10-11

基金项目: 中央高校基本科研业务费专项资金(ZYGX2009J003)

求,论文设计了带有转发域信息树的多分支 Trie 的路由查找算法,以在上述三方面取得平衡.算法依据 Trie 树结构建立了路由表,将最长地址前缀的转发域信息存放在路由表中,实现了快速路由查找功能,同时由路由表中的路由前缀的转发域信息生成了一棵转发域信息树,解决了路由查找中的快速更新问题.

2 路由查找算法分析

为了有效地利用地址空间、降低路由表规模的增长速度,IETF 提出了(Classless Inter Domain Routing, CIDR)^[4]的地址结构,规定可以使用任意长度的地址前缀. CIDR 要求在路由查找时实现最长前缀匹配,在查找过程中不仅需要与目的 IP 的比特值进行匹配,而且需要考虑目的 IP 的掩码长度,所有的路由查找都可以归结为这两方面的匹配查找过程.基于地址长度的路由查找算法的出发点是找到与目的 IP 匹配的掩码长度最长的路由信息.在查找过程中可以使用线性遍历法,也可以使用二分遍历法.

常用的路由查找算法如下:

(1) 基于线性表结构的路由查找算法.线性表结构是最简单的查找数据结构,此算法把所有的路由地址前缀用线性链表的方式组织起来,每一次查找需要遍历线性表中的所有表项,遍历过程中记录最长的路由前缀项,直到遍历完整个链表为止.对于 N 个路由前缀的表项,查找过程的算法复杂度为 $O(N)$,插入路由表项的算法复杂度为 $O(N)$.

(2) 基于二进制 Trie 结构的路由查找算法. Trie 采用一种基于树的数据结构,通过前缀中每一位比特的值来决定树的分支,与路由有关的转发信息就保存在 Trie 树的结点(叶子结点或分支结点).在二进制 Trie 树中进行最长前缀查找的算法就是根据 IP 分组中对应比特的值来选择树的分支,直至到达叶子节点或再无匹配的分支.二进制 Trie 树具有数据结构单一、算法简单、更新容易等优点,但查找过程中需要大量的存储器访问操作.对 IPv4 来说,一颗完整 Trie 树的深度是 32,最坏情况下需要 32 次存储器访问.

(3) 基于前缀长度的二分查找.基于地址长度的二分查找的查找时间与地址长度成对数比例,地址长度越长,其优越性越明显.对于 IPv4 地址来说,最大需要 5 次内存访问操作;但是,二分查找会出现误导的情况,即当某个长度的前缀表中不存在另一

个更长的前缀的子前缀时会出现误导.为解决此问题,Waldvogel^[5]等人提出了加入标记和进行预计算等方法,但这增加了许多额外的空间,而且也使路由表的更新性能变差.

3 带有转发域信息树的 Trie 结构

从前面对现有 IP 路由查找结构及其算法的分析中可以得到如下结论:二分查找具有比较理想的查找速度但更新复杂,而二进制 Trie 树具有算法简单、更新容易的优点,但内存的访问次数过多.带有转发域信息树的 Trie 结构就是借鉴上述几种算法的优点,设计出的兼顾查找速度和更新性能的最长前缀匹配查找的数据结构.

带有转发域信息树的 Trie 结构由两部分组成,一部分是采用固定步长的多分支 Trie 树构成的路由表;一部分是依据目的网段的掩码长度由转发域信息生成的多分支树.一方面,由于采用多分支的 Trie 结构,降低了树的最大深度,提高了查找速率.其中步长可以依据路由器应用环境的不同进行选择,以在提高查找速度的同时提高存储空间利用率;另一方面,因为生成了一棵转发域信息树,对因路由的变化而引起的结点的更新,只是在找到相应的表项或节点后,修改相应的数据,创建或删除转发域信息 Trie 树上的结点,提高了更新性能.

3.1 路由表结构

路由表采用固定步长为 $\{m, n, \dots, n\}$ 的多分支 Trie 树结构,树的最大深度是 $(w-m)/n+1$,其中 w 是掩码长度,最坏情况下只需 $(w-m)/n+1$ 次存储器访问即可找到路由信息.路由表结构如图 1 所示.路由表每个子层均是以路由前缀对应比特位的值为偏移地址的顺序表,例如,路由在第二层的偏移地址就是路由前缀第 $m+1 \sim m+n$ 比特的值.路由的掩码长度小于所在子层能存放的最长前缀值时,就要将路由前缀扩展到多个结点上.例如,向路由表中添加掩码长度为 s 的路由,其中 $m < s < m+n$,在路由表第二层就要占用 2^{m+n-s} 个结点,起始结点的偏移地址为前缀对应比特补零后的值.当 $s = m+n$ 时,起始结点偏移地址即为路由前缀的偏移地址,结点个数为 1.

路由表的每个结点都设计了四个域(字段).这四个域分别是:转发信息域,下一层子表的起始地址域,叶子结点标志域,节点有效标志域.其中下一子层起始地址域和转发信息域共用一块内存,路由表的一个结点只含其中一项.

(1) 转发信息域 *nh*. 该字段指向转发信息树的结点,即下一跳地址. 转发域信息中包含一个引用次数的字段,若此转发域信息被路由表中一个结点指向,或在转发域信息树中成为其他结点的父亲结点,引用次数加 1.

(2) 下一层子表起始地址 *unext*. 如果目的 IP 的掩码长度大于当前层所能存放的掩码长度最大值,则需申请一块能存放包含 2^n 个结点的子表,下一层子表起始地址的指针指向新申请的子表的表头地址.

(3) 叶子结点标志域 *end*. 如果转发域信息在该结点存放,则将叶子结点标志置 1,表示该结点存放有转发域信息,否则置 0. 转发域信息只在叶子结点存放,是路由表中最长地址前缀的转发域信息. 查找路由时,找到了叶子结点即完成了路由查找.

(4) 节点有效标志域 *valid*. 如果路由表结点中含有转发域信息或下一层子表的表头指针时,该字段置 1. 否则置 0,表示从此结点开始的分支中没有存放任何路由信息.

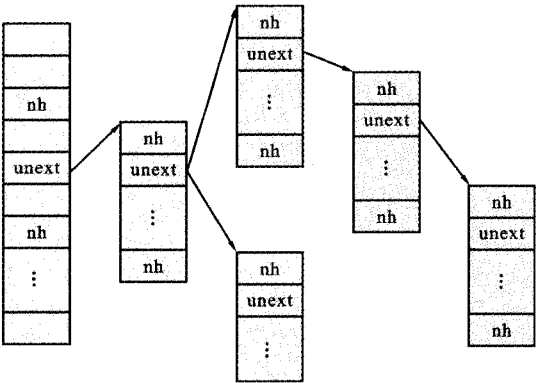


图 1 路由表结构图

3.2 转发域信息树

路由表中的叶子结点只存放最长地址前缀的转发域信息,为便于对路由表进行更新操作,需要记录下具有公共前缀的转发域信息. 因此,依据掩码长度的不同,由添加的路由的转发域信息生成了一棵转发域信息树,树上的结点存放着转发域信息,每个结点的父亲结点就是与其有公共前缀的掩码长度比其更短的结点.

例如向路由表中添加如下路由:10. 192. 200. 26/16、10. 192. 20. 26/20、10. 192. 32. 26/19、10. 192. 16. 20/2、10. 192. 32. 200/23、10. 192. 64. 200/22、10. 192. 16. 200/25、10. 192. 66. 20/27. 这些路由的转发域信息 *nh* 在转发域信息树上表示为 *nh*/maskbits,maskbits 表示路由的掩码长度. 上述路由

生成的转发域信息树如图 2 所示.

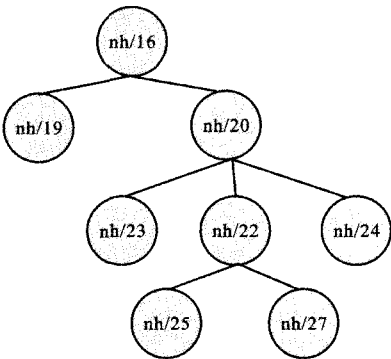


图 2 转发域信息树

4 在带有转发域信息树的路由表中查找最长前缀匹配

4.1 路由查找算法

对于任一目的 IP,路由查找算法是从路由表第一层开始进行查找,逐层判断路由前缀相对路由表每层的偏移的表项,找到叶子结点即完成路由查找,算法流程图如图 3 所示,其中 *level=0* 表示从路由表第一层开始查找.

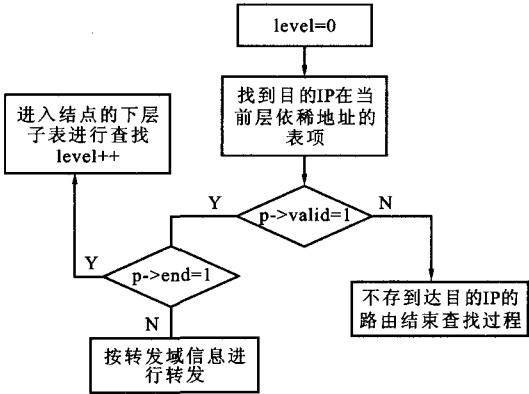


图 3 路由查找算法流程图

4.2 路由添加算法

路由添加算法是从路由表第一层开始进行操作,逐层判断路由前缀 *data* 相对路由表每层的偏移的表项,直至找到能存放路由前缀 *data* 的表项,存放路由信息并将转发域信息 *nh* 连接到转发域信息树上. 算法流程图如图 4 所示. 其中 $p \rightarrow umask$ 表示结点 *p* 中存放路由前缀的长度,maskbits 表示所添加路由前缀的长度,nh-tree 表示转发域信息树.

4.3 删除算法

路由删除算法较为简单,依据查找算法找到要删除的路由前缀的表项或结点后,将其转发域信息修改为结点转发域信息树上的父亲节点,并使结点的转发域信息 *nh* 引用次数减 1.

将来要使用的长度达128比特的IPv6地址,可借鉴二分查找的算法以64、32为分界点构建1棵深度为2的二叉树.在此基础上,再构建步长为 $\{m, n, \dots, n\}$ 多分支Trie树,多分支Trie即为二叉树的叶子结点.这样对于任一IPv6分组,在步长为 $\{16, 4, 4, 4, 4\}$ Trie树结构下,最多情况只需要7次存储器访问次数.

5 结束语

一个好的路由查找算法,应该在查找速度、路由表更新的开销,以及所需存储空间的大小^[6]等方面综合考虑,以取得较佳的综合性能为主要目标.

本文借鉴Trie树的数据结构思想,建立了固定步长的路由表.在此基础上,由添加到路由表中的路由信息依据目的网段和掩码长度,生成了一棵转发域信息Trie树.一方面,由于采用了固定步长的多分支Trie树结构从根本上降低了树的高度,因而就提高了查找速度.另一方面,由于对转发域信息建立了一棵Trie树,在这样的数据结构下,对因路由的变化而引起的结点的更新,只是在找到相应的表项或节点后,修改相应的数据,创建或删除转发域信息Trie树上的结点.因此,更新操作的时间花销仍主要在查找,其算法复杂度与查找算法相当.综上所述,带有转发信息树的多分支Trie结构既保证了查找算法的简单性,又能保证路由信息的易更新性,其综合性能是比较理想的.它是一种具有算法简单、查

找速度快、存储空间利用率较高、便于向IPv6过渡等特点的快速IP路由查找算法.

参考文献:

- [1] 颜顺华,刘宏利.一种改进的端到端混合QoS模型[J].微电子学与计算机,2009,26(5):36-39.
- [2] 刘宝起,王长山,李瑞.基于保证QoS的片上网络路由算法[J].微电子与计算机,2009,26(7):53-56.
- [3] Yeim-Kuan Chang, Yung-Chieh Lin. A fast and memory efficient dynamic IP lookups algorithm based on B-tree [C]// International Conference on Advanced Information Networking and Applications. Bradford: IEEE,2009:278-284.
- [4] Douglas E Comer. Internetworking with TCP/IP Vol I: principle, protocols, and architectures [M]. 4ed. London: Prentice Hall, 2000.
- [5] Waldvogel M, Varghese G, Turner J, et al. Scalable high speed IP routing lookups [J]. Computer Communication Review, 1997,27(4):25-26.
- [6] 谭明锋,高蕾,龚正虎,等. IP路由查找算法研究概述[J]. 计算机工程与科学,2006,28(6):87-91.

作者简介:

邵国良 男,(1985-),硕士研究生.研究方向为无线通信与嵌入式系统.

李广军 男,(1950-),教授,博士生导师.研究面向为通信系统设计、通信专用集成电路设计、无线传感器网络、嵌入式系统设计、EDA/SoC/SOPC设计.

(上接第162页)

5 结束语

虽然语音降噪增强方法的效果很好,但这种方法会带来一定的增强失真,包括增强引起的语音频谱变形和增强后的剩余噪声,它们限制了语音增强对语音识别系统抗噪声性能的进一步提高.针对这些问题,分析了特征空间的倒谱均值归一化方法和模型空间的并行模型合并算法,从识别率上给出了不同补偿技术的补偿效果.

参考文献:

- [1] 王晶,傅丰林,张运伟.语音增强算法综述[J].声学电子工程,2005(1):22-26.
- [2] Ding Qi, Xu Wang, Xu Jinfu, et al. Speech enhancement based on hearing masking properties and subspace [J]. Signal Processing,2008,(1):307-310.

- [3] Ruhi Sarikaya. Robust and efficient technology for speech recognition in noise[J]. Abstracts International, 2009,62(10):4699.
- [4] Oma Y N B, McInnes F R, Jack M A. Improving performance of spectral subtraction in speech recognition using a model for additive noise[J]. Speech and Audio Processing,2008,(11):579-582.
- [5] 王晶,傅丰林,陈建.基于小波变换多阈值语音增强处理研究[J].声学电子工程,2010,(4):32-35.
- [6] 靳雪莲,杨益新.水下环境噪声采集系统的设计与实现[J].微电子学与计算机,2010,27(5):120-123.

作者简介:

苗玉杰 男,(1972-),硕士,讲师.研究方向为信号处理技术、计算机控制工程及算法.

刘雪飞 女,(1972-),硕士,副教授.研究方向为信号处理技术,模式识别.