



基于集合分割的虚拟路由器转发表查找的实现^{*}

胡颖^{1,2}, 庄雷¹

(1. 郑州大学信息工程学院 郑州 450001; 2. 商丘师范学院计算机与信息技术学院 商丘 476000)

摘要: 虚拟路由器即在一个物理路由器上并行实现多个独立的虚拟路由器实例, 是未来互联网的核心网络设备。针对虚拟路由器转发表的查找技术, 采用集合分割的方式, 将分割出的两个集合分别利用 SRAM 和 DRAM 实现。对各种方案进行了分析, 并提出了使用标志位解决 SRAM 查找相交前缀的效率低下问题。实验结果表明, 提出的解决方案大大减少了存储空间, 并在总体上得到了优化。

关键词: 虚拟路由器; 虚拟网; 转发表查找; 集合分割

doi: 10.3969/j.issn.1000-0801.2014.10.016

Realization of Looking up Virtual Router's FIB Based on Set Partition

Hu Ying^{1,2}, Zhuang Lei¹

(1. College of Information Engineering, Zhengzhou University, Zhengzhou 450001, China;

2. College of Computer and Information Technology, Shangqiu Normal University, Shangqiu 476000, China)

Abstract: Virtual router is a core network device of future internet, which runs multiple virtual routers in parallel on a common physical router platform. It is aimed at the looking up technology on virtual router's forwarding table, which using the way of set partition to cut apart the prefixes set into two sets that realized using SRAM and DRAM respectively. Various solutions were analyzed, and new method that to use mark bit solving the problem of inefficient looking up overlapping prefixes in SRAM, was proposed. The experiment results show that the solution greatly reduces storage space, and optimizes the overall effects.

Key words: virtual router, virtual network, forwarding table lookup, set partition

1 引言

虚拟网 (virtual network) 是指在一个实际基底网络 (substrate network) 基础设施上构建多个相互隔离的并行网络切片 (slice), 每个网络切片运行一个具有不同体系结构的虚拟化网络。虚拟网是一种有应用前景的网络结构, 可以在不用改变网络正常操作的情况下, 支持协作、订制路由、新网络协议的实验和部署。基于网络虚拟化的未来互联网技术不仅支持新体系结构、新协议和新算法, 还应

兼容基于 TCP/IP 的互联网。

虚拟路由器 (virtual router) 作为虚拟网的核心网络设备, 近年来引起了众多关注^[1-9]。虚拟路由器是在一个物理路由器平台 (physical router platform) 上并行实现多个相互独立的虚拟路由器实例 (virtual router instance), 每个虚拟路由器实例运行各自的路由协议和转发算法, 并使用各自的转发表 FIB (forwarding information base)。每个物理路由器平台需支持几十个甚至几百个虚拟路由器实例。例如, Cisco 公司和 Juniper 公司的虚拟路由器^[10,11]分别采用虚拟

^{*} 国家重点基础研究发展计划 (“973”计划) 基金资助项目 (No.2012CB315901)



机 VMware 和 Xen, 在通用路由器硬件平台上可实现 128 个虚拟路由器实例。每个虚拟路由器实例对应一个转发表 FIB。在转发表查找技术上, 新的结构带来了新的问题。

(1) FIB 的存储问题

FIB 大小随着网络规模的增大而迅速增大。例如, 现有的大容量 TCAM (ternary content addressable memory, 三态内容寻址存储器) 可容纳 2^{20} 条 40 bit 的路由规则^[12]。如果每个虚拟路由器包含 2^{19} 条路由规则, 且每个路由器维护一个独立的 FIB, 将导致虚拟路由器的存储空间开销呈爆炸性增长, 限制并行虚拟路由器实例的个数。

(2) FIB 的更新问题

目前 Internet 转发表的更新频率为每秒几百次, 更新频率峰值超过每秒 1 万次^[13]。在虚拟路由器中, 一个更新事件会引发对多张转发表的同时更新, 提高了更新频率。为了减少存储, 通常将多张转发表合并成复杂的数据结构, 这给更新操作带来更大的挑战^[14]。

(3) FIB 的查找速度是共性问题

网络链路速率已达到 40 Gbit/s^[15], 因此最长前缀匹配的 IP 地址查找速度成为网速的瓶颈。为使网络达到 40 Gbit/s 的速率, 对于分组长 40 byte 的查找时间就需要达到 8 ns。

尽管最直接的解决方案是为每张转发表分配单独的空间, 这样就满足了虚拟网的隔离性要求, 但实际却很难做到。一方面, 由于不能事先确定各个虚拟路由器实例的转发表大小, 且虚拟路由器实例也是动态变化的, 因此很难确定为每个实例转发表分配多大的空间; 另一方面, 如前面所提到的, 存储空间开销的约束, 使得目前的技术很难同时隔离地维持几十张甚至几百张转发表。

考虑到虚拟路由器中可存在几十张到上百张转发表, 其中有基于 IPv4 的、IPv6 的和非传统路由方式 (如基于 NDN 的名字命名方式) 的转发表, 可将转发表按类型合并存储。本文关注基于 IP 地址的转发表存储。

2 相关研究

目前关于解决转发表的大容量存储问题的研究主要包括以下几点。

- 参考文献[5]提出将每个转发表对应的特里树合并成一个特里树, 来自不同转发表但相同前缀的表项合并为一项, 如果各特里树的结构类似, 即合并的表项居多, 那么这种方案可以达到较好的效果。但

如果各个特里树结构相差较大, 该方案不能达到预期效果。

- 参考文献[11, 12]提出利用特里树编织来解决结构相差很大的问题。该算法对每个特里树节点的左右孩子节点进行旋转, 通过编织比特指示该特里树的遍历方向, 从而确保查找正确。这样, 利用孩子节点旋转机制增加多个 FIB 的重叠度, 进一步减少融合特里树的存储空间开销。Ganegedara 在参考文献[16]中提到, 在某些情况下, 使用特里树编织也很难有效地合并转发表, 比如同属一个区域的边缘路由器转发表中大部分前缀是相同的, 而来自不同边缘路由器的转发表相差就很大。于是 Ganegedara 提出了多根 (multiroot) 的解决办法, 不同区域且相差很多的转发表分别使用各自的特里树。还可以结合特里树编织技术, 减小特里树结构差距大所带来的影响。本文假设各个转发表是相似的, 实际环境中, 若不相似, 可采用多根技术或特里树编织技术。上述研究主要关注高效 IP 地址查找的存储, 由于多 FIB 融合的更新开销高, 如何在存储高效的条件下支持快速更新成为快速 IP 地址查找算法的关键问题。
- 参考文献[1, 2, 4]均是基于集合分割的 IP 地址查找算法。集合分割借助特里树将前缀集合分割成两个, 其中不相交的集合大约包括 90% 的前缀; 另一个重叠前缀的集合包括大约 10% 的前缀。这样, 大部分的更新操作只涉及不相交前缀集合, 大大降低了更新速度。
- 参考文献[4]采用并行的流水线结构, 并采用 2-3 树数据结构表示每个子集, 与 TCAM 相比, 查找和更新速度较慢。
- 参考文献[1]对于两个子集分别采用 TCAM 技术和 SRAM (static RAM, 静态随机存储器) 技术, 并使用虚拟路由器身份 (virtual router ID, VID) 的方式合并转发表, 使得合并转发表的长度和转发表的个数线性相关。当需要合并上百张转发表时, 占用的存储空间将是相当大的。
- 参考文献[2]采用 TCAM 分别存储两个子集, 对于重叠前缀集合, 采用 VID 的方式合并转发表。尽管文中对于 14 个转发表的情况进行了分析, 结论是可行的, 但虚拟路由器可能承载几百个虚拟网, 这种合并方式会占用过多的存储量。另外, TCAM 的价

格和功耗都是很高的,而且查找和更新操作大都出现在不相交前缀集合,使用 TCAM 存储重叠前缀集合的更新开销也较大,因此,对于重叠前缀集合,使用 TCAM 的必要性不大。

3 相关技术

3.1 基于特里树的集合分割

集合分割是指将一个融合的 FIB 映射到一棵特里树,将所有叶子节点对应的前缀组成一个集合,这个集合中所有前缀不相交,称为不相交前缀集合;将内部节点对应的所有前缀组成一个集合,不能保证这个集合中的所有节点不相交,称为重叠前缀集。

实验结果表明,叶子节点约占 90%,内部节点约占 10%。路由更新更多地体现在长前缀的增加或者删除上,短的内部节点前缀相对稳定。

3.2 TCAM 和 SRAM

基于 TCAM 和基于 SRAM 的 FIB 查找技术都是常用的 FIB 查找技术,它们采用不同的方法。

在 TCAM 中,每个前缀与 IP 地址并行比较,并在一个时钟周期完成。如果前缀集合是不相交的,那么每次最多有一个匹配项;如果前缀集合是重叠的,有可能一次匹配多个,通常是将前缀按长度排序,如较长的前缀存放在低地址,那么在匹配前缀中选取低地址前缀作为匹配结果。这样,在更新时,为了保持顺序,需要移动大量不相关的前缀项。

SRAM 通常使用树型数据结构。但是,基于特里树的结构通常存储量较大,更新开销也较大。且由于前缀重叠,通常的遍历树算法不可用,FIB 查找可能需要多次访问内存(以下简称访存)。对于较大的转发表,存储空间的限制成为主要约束。

相比传统需要多次访存的基于软件的路由查找操作来说,TCAM 在一个时钟周期内可以完成一次路由查找,因此具有一定的时间性能优势。但是,TCAM 的造价和功耗较大。

已知集合分割后,不相交前缀占 90%,如果将其用 TCAM 存储,查找快,占用存储量较使用 SRAM 要小,并且由于是不相交的前缀集合,不需要保持顺序存储,更新时仅需要一次访存操作,大大简化了更新,因此,与参考文献[15]和参考文献[17]的做法一样,这里使用 TCAM 存储不相交前缀集合。

重叠前缀集合占 10%,如果使用 TCAM,那么为保持顺序更新,需要多次访存。针对 TCAM 增量更新问题,有许多解决方案,常被采纳的有参考文献[18]的两个算法: PLO_OPT 和 CAO_OPT 算法。但是,这些算法只是在一定程度上优化了更新性能。SRAM 耗费存储空间较大,查找需要多次访存,速度较 TCAM 慢,但由于使用新一代的 FPGA,可以使得更新和查找操作同时进行,互不干扰,因此,这里使用 SRAM 存储重叠前缀集合。

3.3 转发表合并

转发表合并存储可以采用两种方式:参考文献[19]的合并方式记为合并方式 1;参考文献[20]的合并方式记为合并方式 2。合并方式 1 中,给所有转发表等长编号,将转发表编号和前缀合并标识每个前缀,将转发表纵向延伸;合并方式 2 中,将每张转发表中前缀值相同的表项合并(如果相应转发表中没有对应前缀,可以将下一跳地址标识为 0),将转发表横向延伸。如对于转发表 A 的表项 $\langle 0^*, B \rangle$ 和转发表 B 的表项 $\langle 0^*, C \rangle$,可以合并为 $\langle 0^*, [B, C] \rangle$ 。

参考文献[15]在 TCAM 和 SRAM 中合并转发表时均采用合并方式 1,这样就不支持个数较多的转发表,这里假定有 100 张转发表,每个转发表总记录数为 500 000 个,对每个转发表划分出的不相交前缀个数大约为 400 000 个,容量较大的 TCAM 可容纳 2^{20} 个 40 bit 的前缀,若采用合并方式 1,容纳重叠前缀就需要大约 39 ($400\,000 \times 100 / 2^{20} \approx 38.15$) 个 TCAM 芯片,这是不可行的;若采用合并方式 2,由于可以合并多个转发表的记录,以减少 TCAM 的使用,因此本文采用合并方式 2 来合并转发表。

4 总体结构

图 1(a)显示了将两张转发表合并的结果,图 1(b)是合并转发表对应的 1 bit 特里树。

和参考文献[15]的结构类似,如图 2 所示,这里有两个并行的 IP 地址查找引擎。其中,较大的集合映射到基于 TCAM 的查找引擎,较小的集合映射到基于 SRAM 的查找引擎。分组头解析器(head parser)将到来分组中的分组头部分分离,提取出目的 IP 地址,并将此 IP 地址传到并行结构,使得它们同时启动查找。同时,到来的分组缓存在 buffer 中,等待查找结果。两个并行结构分别产生下一跳信息,由于 TCAM 中存储较长的前缀,所以优先权仲裁器(priority arbiter)优先使用 TCAM 产生的结果。由于两个引

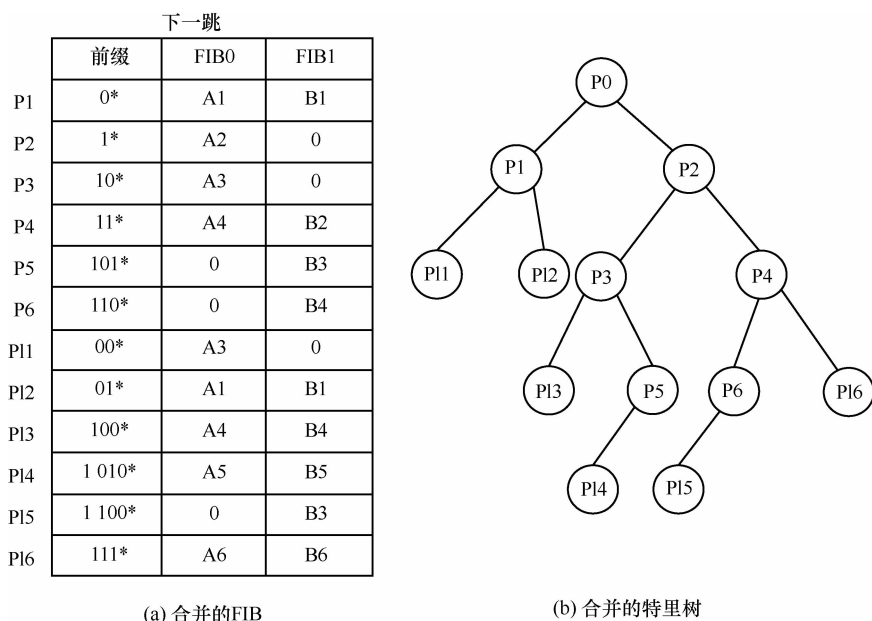


图1 转发表合并结果及其对应的特里树

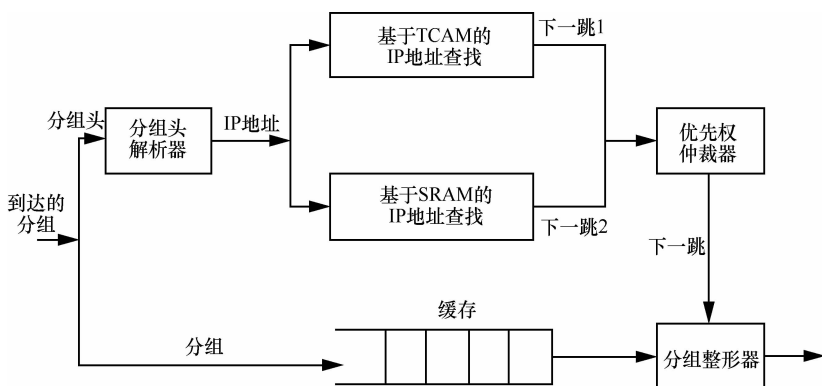


图2 总体结构

擎都有可能匹配不到结果,优先权仲裁器决定是否丢弃或最终的下一跳。将 buffer 中暂存的分组取出,并根据下一跳信息,由分组整形器(packet modifier)对分组整形,并将分组送到相应的输出接口。

本文的 TCAM 和 SRAM 结构如图 3(a)、图 3(b)所示。

由于已经假设各转发表数据集相似,所以基于 TCAM 的存储结构可以采用合并方式 2。这种结构会降低命中率,因此如果没有前面的假设,应对转发表进行预处理(如采用多根技术),使得为 0 的表项较少。

对于基于 SRAM 的查找结构,这里采用合并方式 2 合并转发表,查找中出现了新的问题。已知对特里树查找需要从根遍历到叶子节点或将前缀读完,这个过程中一旦遇到匹配节点,需要记录节点信息,最终返回的是最后一次匹配的节点。如果采用合并方式 1,一个节点最多指向一

个转发表项,只要节点有指向下一级 SRAM 的指针,就说明该节点标识了一个前缀,记录节点信息时不需要到下一级 SRAM 中查看该表项是否存在;若采用合并方式 2,一个节点指向多个转发表项(这里是 13 个),有些表项可能不存在(图 3 中标识为 0),因此每匹配一个节点需要到下级 SRAM 查看,降低了查找效率。针对这个问题,在特里树节点增加了标志位,0 或 1 分别代表无或有表项,这样每次查找仅需要最后一次访问下级 SRAM,如图 4 所示。

5 快速增量更新

总的来讲,快速增量更新分为 3 类:插入、删除、更改。更改操作不需要改变特里树的结构,这里着重讨论插入和删除操作。

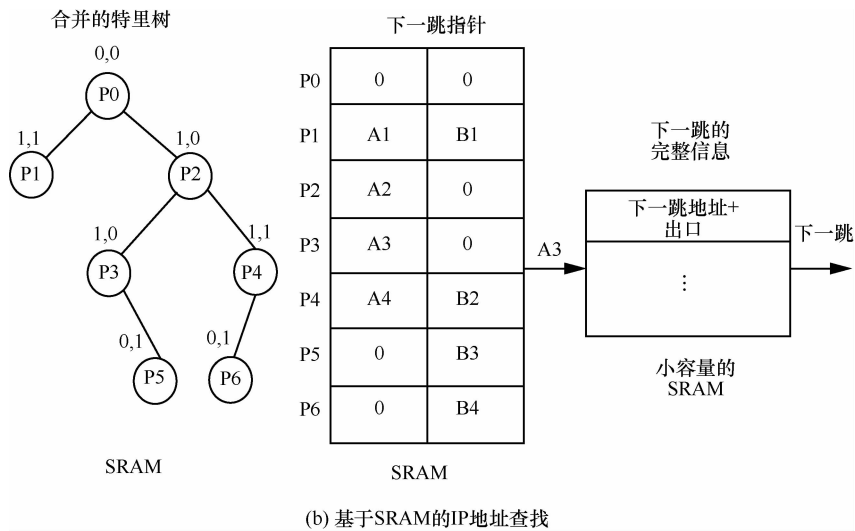
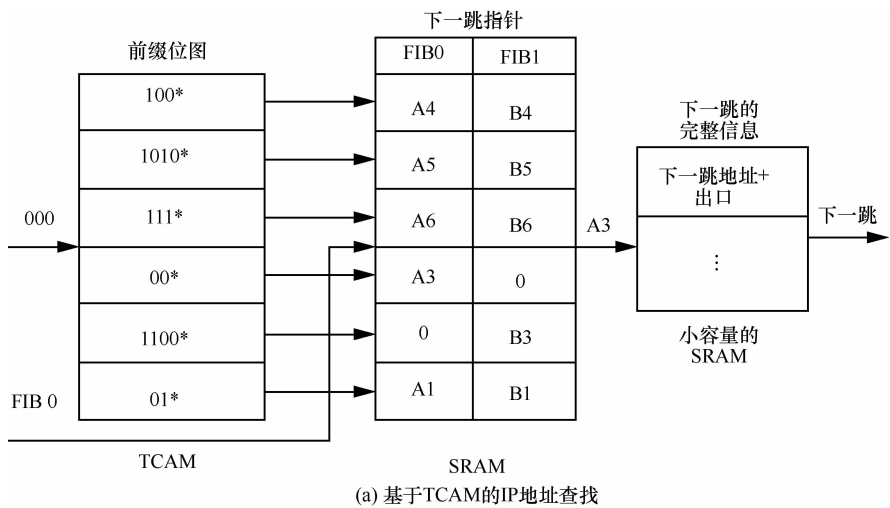


图 3 基于 TCAM 和基于 SRAM 的 IP 地址查找

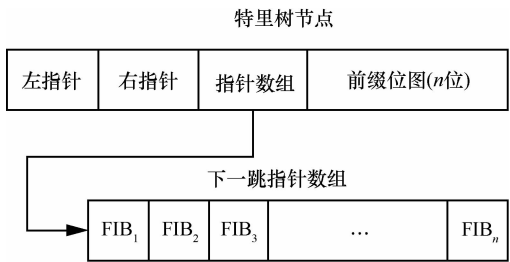


图 4 特里树节点和下一跳指针数组的数据结构

更新操作分两步进行:控制面确定在两个集合中插入或删除哪些项;数据面执行确切的插入或删除项的操作。对于第一步,控制面维持一棵 1 bit 特里树,分割不相交前缀集合(S1)和重叠前缀集合(S2)。方案 1、方案 2 和本文的方案是一样的。在第二步中,执行对 S1 或 S2 的插入或删除项操作。其中,S1 是不相交前缀集合,不需要保证顺序,

因此,对 S1 插入或删除只需要一次写操作即可。S2 算法的具体过程如下。

输入:更新前合并转发表 F_0 ,待插入(删除)记录的前缀值 P 、所属转发表编号 F 及对应下一条值 N 。

输出:更新后合并转发表 F_0' 。

插入过程为:如果 P 在 F_0 中,将 P 和 F 对应项的值改为 N ,将 P 和 F 对应标记项改为 1;否则,在 F_0 中插入记录,其前缀项的值为 P , P 和 F 对应项的值为 N , P 和 F 对应标记项为 1,该记录中其余项置为 0。

删除过程为:将 P 和 F 对应项置为 0,将 P 和 F 对应标记项置为 0,如果该记录对应所有标记项都为 0,删除该条记录。

6 实验与分析

本次实验数据是在项目 RIS 中收集的 2013 年 9 月 12 日



表 1 对实际转发表的分析

FIB	总 IPv4 前缀数	特里树节点数	叶子前缀数	剪枝后前缀数	剪枝后节点数
Rrc00	475 101	1 124 306	428 562	46 539	470 698
Rrc01	465 890	1 098 720	420 138	45 752	458 428
Rrc03	479 201	1 132 898	432 333	46 868	473 650
Rrc04	475 101	1 124 306	428 562	46 539	470 698
Rrc05	467 751	1 103 188	421 906	45 845	460 172
Rrc06	473 920	1 119 093	427 144	46 776	467 970
Rrc07	468 235	1 105 746	422 572	45 663	461 627
Rrc10	473 264	1 124 407	426 899	46 365	473 143
Rrc11	475 567	1 122 722	429 088	46 479	468 522
Rrc12	482 544	1 144 723	434 829	47 715	481 489
Rrc13	512 776	1 215 634	464 831	47 945	504 693
Rrc14	476 914	1 123 758	429 761	47 153	468 842
Rrc15	476 057	1 122 188	428 316	47 741	469 755
本 FIB	541 921	1 291 949	488 900	53 021	543 916
参考文献[15]	6 202 321	14 661 716	5 594 941	607 380	6 129 714
参考文献[17]	541 921	1 291 949	488 900	607 380	6 129 714

16 时的数据,13 个路由转发表的前缀总数以及叶子节点对应前缀数见表 1。采用本文提出的合并路由转发表的方法,将 13 个路由转发表合并为一个,并将其分解成两个集合。

对于参考文献[15]和参考文献[17]所采用的合并方式1,原始前缀前附加代表转发表的前缀均为 4 位(0000-1100)。参考文献[15]中的存储量减少明显,而参考文献[17]在合并 13 张转发表时存储量是可以的,但当转发表数量达到几十张甚至几百张时,存储量将大大增加,这样无论是 TCAM 的数量,还是更新效率,都不是理想的效果。本文的合并 FIB 方式,大大减少了存储空间。

7 讨论

TCAM 中,对于合并后转发表的每个前缀,不是所有原转发表都有对应前缀项(也就是标识为 0 的项)。造成这种现象的根源在于,控制面构造特里树时,其他转发表的前缀可能覆盖原转发表的前缀,使得被覆盖的前缀不再是叶子节点。因此,这种结构会增加不命中的概率。于是,当各前缀集合相差较大时,适宜使用合并方式 1;当相差不大时,采用合并方式 2。

若将重叠前缀集合变为不相交的,常用叶推(leaf-pushing)技术^[20]。它存在两个缺陷:第一,实施叶推技术会产生冗余节点,使节点个数增加 6%左右,相对总的前缀个数而言,增量并不是很大(前缀个数增加大约 0.6%);第二,每次更

新(如插入或删除)操作,可能需要重建叶推特里树,并需要和原叶推特里树对比,从而得到需要改动的多个前缀项。这样,不管在控制面还是数据面(尤其是控制面),需要耗费较多的空间和时间。这是致命的缺陷,因此这里不采用叶推技术处理重叠前缀集合。

多比特特里树可以有效缩短查找时间,但是更新开销很大。而在虚拟路由器中,一个更新事件会引发对多张转发表的同时更新,更新频率增加很多,所以这里采用 1 bit 特里树的结构存储。

8 结束语

本文分析了基于集合分割的虚拟路由器转发表的实现,对几种可能采取的技术进行了阐述,并采用了合并方式 2 来合并两个集合,实验结果表明,这种合并方式大大减少了存储空间。并针对合并方式 2 带来的弊端,在一级 SRAM 中使用标志位,使得在查找过程中访问下级 SRAM 的次数降为 1 次。

参考文献

- 1 Anderson T, Peterson L, Shenker S, *et al.* Overcoming the internet impasse through virtualization. *Computer*, 2005, 38(4): 34-41
- 2 Turner J S, Taylor D E. Diversifying the internet. *Proceedings of Global Telecommunications Conference*, St Louis, MO, USA, 2005

- 3 Feamster N, Gao L, Rexford J. How to lease the internet in your spare time. *ACM SIGCOMM Computer Communication Review*, 2007, 37(1): 61~64
- 4 FIRE. <http://www.ict-firworks.eu>, 2012
- 5 Zhang L, Estrin D, Burke J, *et al.* Named Data Networking (NDN) Project. Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC, 2010
- 6 Egi N, Greenhalgh A, Handley M, *et al.* Towards high performance virtual routers on commodity hardware. *Proceedings of the ACM CoNEXT Conference*, New York, NY, USA, 2008:1~12
- 7 NWGN. <http://akari-project.nict.go.jp>, 2012
- 8 Han S, Jang K, Park K S, *et al.* PacketShader: a GPU-accelerated software router. *ACM SIGCOMM Computer Communication Review*, 2010, 40(4): 195~206
- 9 FIF. <http://www.fif.kr>, 2012
- 10 Cisco logical router. <http://www.cisco.com>, 2012
- 11 Juniper logical router. <http://www.juniper.net>, 2012
- 12 Netlogic. NL 9000 RA knowledge-based processors, 2009
- 13 The BGP instability report. <http://bgpupdates.potaroo.net/instability/bgpupd.html>, 2014
- 14 Huang K, Xie G, Li Y, *et al.* Offset addressing approach to memory-efficient IP address lookup. *Proceedings of IEEE INFOCOM*, Shanghai, China, 2011: 306~310
- 15 Luo L, Xie G, Xie Y, *et al.* A hybrid IP lookup architecture with fast updates. *Proceedings of IEEE INFOCOM*, Orlando, FL, USA, 2012: 2435~2443
- 16 Ganegedara T, Jiang W, Prasanna V. Multirout: towards memory-efficient router virtualization. *Proceedings of IEEE International Conference*, Kyoto, Japan, 2011: 1~5
- 17 Luo L, Xie G, Uhlig S, *et al.* Towards TCAM-based scalable virtual routers. *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, Nice, France, 2012: 73~84
- 18 Shah D, Gupta P. Fast incremental updates on ternary-CAMs for routing lookups and packet classification. *Proceedings of Hot Interconnects-8*, Stanford, CA, USA, 2000: 145~153
- 19 Fu J, Rexford J. Efficient IP-address lookup with a shared forwarding table for multiple virtual routers. *Proceedings of the ACM CoNEXT Conference*, New York, NY, USA, 2008: 1~12
- 20 Le H, Ganegedara T, Prasanna V K. Memory-efficient and scalable virtual routers using FPGA. *Proceedings of the 19th ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, Monterey, California, USA, 2011: 257~266

[作者简介]



胡颖,女,商丘师范学院讲师,主要研究方向为网络虚拟化。

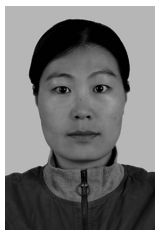
庄雷,女,博士,郑州大学教授,主要研究方向为可重构网络和网络虚拟化。

(收稿日期:2014-08-20)

(上接第 102 页)

- 3 Zhu S, Ding Z. Joint synchronization and localization using TOAs: a linearization based WLS solution. *IEEE Journal on Selected Areas in Communications*, 2010, 28(7): 1017~1025
- 4 Zheng J, Wu Y C. Joint time synchronization and localization of an unknown node in wireless sensor networks. *IEEE Transactions on Signal Processing*, 2010, 58(3): 1309~1320
- 5 Moon S B, Skelley P, Towsley D. Estimation and removal of clock skew from network delay measurements. *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '99)*, New York, USA, 1999:227~234
- 6 Jeske D R. On the maximum likelihood estimation of clock offset. *IEEE Transactions on Communication*, 2005, 53 (1): 53~54
- 7 Johnson N L, Kemp A W, Kotz S. *Univariate Discrete Distributions*. Hoboken: John Wiley & Sons, 2005

[作者简介]



赵响,女,西安电子科技大学博士生,主要研究方向为无线传感器网络中的定位与同步。



林基明,男,桂林电子科技大学教授,主要研究方向为移动通信、无线传感器网络中的定位与同步。

(收稿日期:2014-05-28)