

文章编号:1001-9081(2009)11-2969-03

一种基于规则分解映射的防火墙规则匹配算法

唐晔

(遵义师范学院 计算机科学系,贵州 遵义 563002)

(artherzy@163.com)

摘要:并行树搜索(PTS)算法是报文分类领域中较为优秀的算法之一,但它需要构建大量的 external nodes,且只支持以前缀形式表示的规则,因此其匹配效率及适用范围都受到了很大的影响。针对这一问题,提出一种基于规则分解映射的规则匹配算法 RMBRDM。RMBRDM 算法首先按照启发式方法选取标准维;然后根据规则分解映射和标准维对相关规则进行分解;最后建立一棵二叉决策树。理论分析和仿真实验均表明,RMBRDM 算法不仅支持以范围形式表示的规则,且时空性能优于 PTS 算法。

关键词:规则匹配;并行树搜索算法;平衡二叉决策树

中图分类号: TP309;TP393.08 文献标志码:A

Rule matching mapping algorithm for firewall based on rule decompositon mapping

TANG Ye

(Department of Computer Science, Zunyi Normal College, Zunyi Guizhou 563002, China)

Abstract: Parallel Tree Search (PTS) is one of the best algorithms among the existing algorithms for rule matching. However, PTS needs to construct so many external nodes and only supports rules with prefixes. The authors proposed an algorithm named RMBRDM for rule matching based on rule decomposing. At first, RMBRDM employed heuristic methods to choose a standard dimension. And then rules could be decomposed according to rule decomposing mapping and the standard dimension. At last, a binary decision tree could be built. Algorithm analysis and simulation results show that RMBRDM can support rules with ranges and the performance of RMBRDM is better than that of PTS.

Key words: rule matching; Parallel Tree Search (PTS) algorithm; balanced binary tree

0 引言

近年来,随着 Internet 的不断发展,包括防火墙、路由器在内的许多网络设备都需要支持 QoS,即为不同的流提供不同的服务质量保证。在这种情况下,规则匹配(即报文分类)已经成为了这些网络设备的基本功能之一。规则匹配的基本任务是:当接收到数据包时,搜索预先设置的规则集,找出数据包所能匹配的规则,并按规则定义的动作处理数据包。对于防火墙而言,规则定义的动作通常是放行或者丢弃。一个数据包有时能同时匹配两条或两条以上的规则,且规则间的动作互不一致,这种情况称为规则冲突。常见的解决方案是给不同的规则赋予不同的优先级。通常,规则在规则集中的位置代表了它的优先级。

随着链路速度的提高,特别是随着规则个数的增多,规则匹配已经成为许多网络设备的性能瓶颈,这引起了研究人员的广泛关注,并有不少规则匹配算法被提出^[1]。

现有的众多规则匹配算法可以分为以下几类:基于 TCAM (Ternary Content Addressable Memory) 的匹配算法^[2-3], 基于哈希表的匹配算法^[4-5], 基于决策树的匹配算法^[6-7], 基于 Trie 的匹配算法^[1,8-9], 以及基于分治思想的匹配算法^[10-11]。

常见的基于 TCAM 的匹配算法的优点是匹配速度快,时间复杂度通常为常数。但由于只支持以前缀形式表示的规

则,能量消耗过大,价格高昂,TCAM 的使用范围受到了很大的限制。

对于基于哈希表的匹配算法而言,这类算法通常能提供良好的最坏情况下的性能保证,但其平均性能往往较差。而基于决策树的匹配算法,正好相反。这类算法往往不能提供良好的最坏情况下的性能保证,但其平均性能通常较好。另外,基于决策树的匹配算法还有一个严重的缺点,即需要维护一棵庞大的决策树,空间开销过大。

基于 Trie 的匹配算法,通常只支持以前缀形式表示的规则集,且不能提供良好的最坏情况下的性能保证。而基于分治思想的匹配算法,要么时间复杂度较大,要么空间使用量较大。这类算法通常只适用于中小规模的规则集。

在众多的规则匹配算法中,文献[12]提出的并行树搜索(Parallel Tree Search, PTS)算法是较为优秀的算法之一。PTS 采用文献[9]提出的二阶段匹配机制:第一阶段对源 IP 地址前缀和目的 IP 地址前缀组合进行匹配;在第二阶段,根据第一阶段的匹配结果,顺序搜索数量有限的规则,以确定数据包能匹配的优先级最高的规则。PTS 算法的主要工作在于对第一阶段进行改进,将源 IP 地址前缀和目的 IP 地址前缀组合组织成一棵平衡二叉树,以加快对数据包的处理。

虽然 PTS 算法在一定程度上对文献[9]算法进行了改进,但是 PTS 算法仍然存在以下两方面的问题:1) 在构建平衡二叉树的过程中,PTS 算法需要建立大量的 external nodes,

这不仅影响规则匹配算法的预处理时间,增加了空间使用量,而且降低了数据包匹配效率;2)PTS 算法只支持源 IP 地址分量和目的 IP 地址分量以前缀形式表示,而不支持它们以范围形式表示。由文献[1]可知,一个以范围形式表示的 d 维规则,最坏情况下,将转换成 $(2w - 2)^d$ 条以前缀形式表示的规则。其中, w 是规则分量的域长。例如,IPv4 的 IP 地址分量,其 w 等于 32。

针对 PTS 算法存在的这两个问题,本文提出一种基于规则分解映射的规则匹配算法(Rule Matching Based on Rule Decomposing Mapping, RMBRDM)。RMBRDM 算法首先按照启发式方法选取标准维,再根据规则分解映射和选取的标准维对相关规则进行分解,最后,建立一棵平衡二叉决策树。理论分析和仿真实验均表明,RMBRDM 算法不仅能直接支持以范围形式表示的规则,而且时空效率均优于 PTS 算法。

目前规则的表示形式主要有:确切值表示方式、前缀表示方式和范围表示方式。其中范围表示方式是最一般的表示方式,应用非常广泛。而对于 1.2.3.4/0.255.255.255 这种表示方式,在防火墙规则配置方面应用很少,常见的报文分类算法通常都不考虑这种情况。因此,本文算法也不处理这种表示方式。

1 标准维的选取

与 PTS 算法类似,RMBRDM 算法也采用二阶段方式处理。由于第二个阶段的处理方法与文献[9]算法以及 PTS 算法相同,因此本文不再赘述,而将重点放在第一阶段处理上,即仅讨论源 IP 地址分量、目的 IP 地址分量组合的匹配情况。下面,首先定义几个需使用的符号。

定义 1 $F[i]$ 表示规则集的第 i 条规则, $F[i][j]$ 表示规则 $F[i]$ 的第 j 维分量, $F[i][j][L]$ 表示 $F[i][j]$ 的左端点值, $F[i][j][R]$ 表示 $F[i][j]$ 的右端点值。其中 $i, j \in N^+$ 。 $\text{Priority}(F[i])$ 表示 $F[i]$ 的优先级, $\text{Action}(F[i])$ 表示 $F[i]$ 的处理动作。

由于本文仅讨论源 IP 地址分量和目的 IP 地址分量的组合,因此这里所指的规则集是去除其他分量后形成的规则集。规则集形成方法可见文献[9,12]。因此,任意规则 $F[i]$ 只有两个分量。从文献[1]可知,规则相当于多维空间中的超长方形,规则分量相当于数轴上的线段,而数据包则相当于多维空间中的点。根据这种对应关系,下面给出规则分量间的关系分类。

定义 2 规则分量 $F[i][k]$ 和规则分量 $F[j][k]$, 其中 $i, j, k \in N^+$ 。若 $F[i][k][R] < F[j][k][L]$ 或者 $F[j][k][R] < F[i][k][L]$, 则称 $F[i][k]$ 和 $F[j][k]$ 无关联, 可记为 $F[i][k] \sim F[j][k]$ 。否则, 称 $F[i][k]$ 和 $F[j][k]$ 有关联, 可记为 $F[i][k] \cong F[j][k]$ 。

定义 3 规则 $F[i]$ 和规则 $F[j]$, 称 $F[i]$ 和 $F[j]$ 在第 k 维上相关, 记为 $F[i] \Theta_k F[j]$, 当且仅当 $F[i][k] \cong F[j][k]$ 。

定义 4 与规则 $F[i]$ 在第 k 维上相关的规则个数, 记为 $\text{Number}_k(F[i])$ 。

下面根据上述定义,给出标准维的选取方法,即定义 5。

定义 5 若 $\exists k, k \in N^+$, 使得:

$$\sum_{i=1}^n \text{Number}_k(F[i]) \quad (1)$$

达到最小,则称 k 是 RMBRDM 算法选取的标准维。其中, n 是规则集包含的规则个数。

2 规则分解映射

定义 5 给出了 RMBRDM 算法选取标准维的方法,本节将重点讨论如何对在标准维上相关的规则进行分解。

$$\begin{aligned} TT(t_1, t_2) = & \{t_1\} \cup \{[t_2[L], t_1[R]]\} \cup \{[t_1[R] + 1, t_2[R]]\}, \\ & t_1[L] < t_2[L] \leq t_1[R] < t_2[R] \\ \{t_2\} \cup & \{[t_1[L], t_2[R]]\} \cup \{[t_2[R] + 1, t_1[R]]\}, \\ & t_2[L] < t_1[L] \leq t_2[R] < t_1[R] \\ \{t_1\} \cup & \{[t_2[L], t_1[L] - 1]\} \cup \{[t_1[L], t_1[R]]\} \cup \\ & \{[t_1[R] + 1, t_2[R]]\}, \\ & t_2[L] < t_1[L] \leq t_1[R] < t_2[R] \\ \{t_1\} \cup & \{[t_2[L], t_1[R]]\} \cup \{[t_1[R] + 1, t_2[R]]\}, \\ & t_2[L] = t_1[L] \leq t_1[R] < t_2[R] \\ \{t_1\} \cup & \{[t_2[L], t_1[L] - 1]\} \cup \{[t_1[L], t_2[R]]\}, \\ & t_2[L] < t_1[L] \leq t_1[R] = t_2[R] \\ \{t_1\} \cup & \{t_2\}, \\ & t_2[L] = t_2[L] \leq t_1[R] = t_2[R] \\ \{t_2\} \cup & \{[t_1[L], t_2[L] - 1]\} \cup \{[t_2[L] + 1, \\ & t_2[R]]\} \cup \{[t_2[R] + 1, t_1[R]]\}, \\ & t_1[L] < t_2[L] \leq t_2[R] < t_1[R] \\ \{t_2\} \cup & \{[t_1[L], t_2[R]]\} \cup \{[t_2[R] + 1, t_1[R]]\}, \\ & t_1[L] = t_2[L] \leq t_2[R] < t_1[R] \\ \{t_2\} \cup & \{[t_1[L], t_2[L] - 1]\} \cup \{[t_2[L], t_1[R]]\}, \\ & t_1[L] < t_2[L] \leq t_2[R] = t_1[R] \\ \emptyset, & \text{其他} \end{aligned} \quad (2)$$

规则的分解以两条规则为基本的处理单位,根据选取的标准维分解相应的规则。因此,下面仅以两条规则为讨论对象分析规则分解映射。

规则分量分解映射定义如下。

定义 6 规则分量分解映射 $TT: T \times T \rightarrow 2^T$, T 是所有可能的规则分量组成的集合。 $T[L]$ 代表左端点, $T[R]$ 代表右端点。 $\forall t_1, t_2 \in T$, 规则分量分解映射如式(2)所示。

根据定义 6 可以获得规则分解映射,即定义 7。

定义 7 k 是 RMBRDM 算法选取的标准维, $k \in N^+$ 。规则分解映射 $RR: R \times R \rightarrow 2^R$, R 是所有可能的规则组成的集合。规则 $F[i], F[j] \in R$:

$$RR(F[i], F[j]) = \begin{cases} \bigcup_{q=p}^{p+num} \{F[(q)]\}, & F[i] \Theta_k F[j] \\ \emptyset, & \text{其他} \end{cases} \quad (3)$$

其中 $p, q \in N^+, p$ 是规则集可用下标起始值, $num = |TT(F[i][k], F[j][k])| - 1$ 。 $\forall r, r \neq k, r \in N^+$, 均有 $F[q][r] = F[j][r]$, 而 $F[q][k] \in TT(F[i][k], F[j][k])$ 。

经过规则分解映射处理后,任意两条规则的标准维分量要么是无关联的,要么是相等的,即定理 1。

定理 1 k 是 RMBRDM 算法选取的标准维, $k \in N^+$ 。经过规则分解映射处理后,规则 $F[i]$ 和 $F[j]$, 均有 $F[i][k] \sim F[j][k]$, 或者 $F[i][k] = F[j][k]$ 。

证明 假设 $F[i][k][L] < F[j][k][L] < F[j][k][R] < F[i][k][R]$ 。从式(2)、(3)可知,这种分量关系将被分解成:

$$\{[F[i][k][L], F[j][k][L] - 1]\} \cup \{F[j][k]\} \cup \{[F[j][k][L], F[j][k][R]]\} \cup \{[F[j][k][R] + 1, F[i][k][R]]\}$$

显然,这些分量要么是相等的,要么是无关联的。对于分量的其他关系,同理可证。

所以,定理 1 成立。

经过规则分解映射处理后,规则间还是可能存在冲突,这时可以按照常见的冲突消除算法消除这些冲突规则。

3 决策树的建立

经过规则分解映射处理后,任意两条规则的第 k 维(假设 k 是标准维),要么是相等的,要么是无关联的。根据这一性质,RMBRDM 算法按照第 k 维分量,将规则组织成一棵二叉决策树。通常组织二叉树的关键在于确定树节点(这里即规则)的大小顺序关系。规则的大小关系定义如下:

规则 $F[i]$ 和规则 $F[j]$,若 $F[i][k][R] < F[j][k][L]$,或者, $F[i][k] = F[j][k]$ 但 $F[i][r][R] < F[j][r][L]$,则 $F[i] < F[j]$,其中 $r \neq k$ 。对于其他情况,均有 $F[j] < F[i]$ 。

RMBRDM 算法在实际实现中采用红黑树的方式将规则组织成一棵平衡二叉树。

数据包的匹配过程就是对二叉决策树进行搜索。由于数据包相当于多维空间中的一个点,因此,它可以被当作一个特殊的规则。也就是说,数据包的搜索方法即二叉决策树的组织方法,因此,不再赘述。

4 算法分析与仿真实验

本文对 RMBRDM 算法和 PTS 算法进行了对比实验。实验的规则集是按照文献[5]中介绍的综合方法随机生成的规则集。该方法随机生成的规则集依赖于母规则集,而母规则集选取自某公司商业防火墙。因此,测试所用规则集主要适合于常见的防火墙应用。随机生成的规则集包含的规则数目 100~5 000 不等。实验环境:Intel P4 2.8 GHz,1 GB 内存。下面从数据结构建立、数据包匹配的时间性能、空间性能等三方面进行讨论。

1) 图 1 描述了 RMBRDM 算法和 PTS 算法在构造数据结构过程中所消耗时间的关系。在图 1 中,横坐标表示规则个数,纵坐标表示 PTS 算法平均构造时间和 RMBRDM 算法平均构造时间的比值。

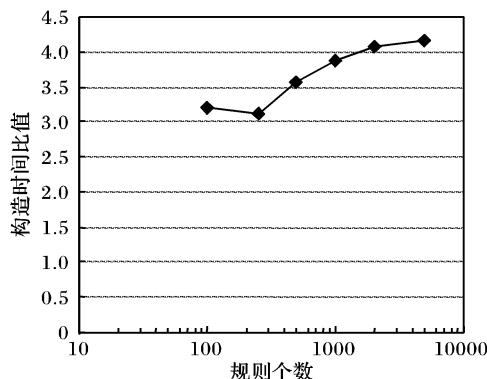


图 1 构造时间对比

从图 1 可知,RMBRDM 算法构造时间少于 PTS 算法。PTS 算法构建二叉树所需时间基本是 RMBRDM 算法构建时间的三四倍之多。造成这种情况主要有以下两方面原因:一是 PTS 算法需要花大量时间构建 external nodes,而 RMBRDM 算法完全不需要建立 external nodes;二是由于实验所用规则

是以范围形式表示的,因此 PTS 算法需要先将其转换成前缀形式。如前所述,一个以范围形式表示的 d 维规则,最坏情况下,将转换成 $(2w - 2)d$ 条以前缀形式表示的规则。在本文所用的实验规则集中,规则集转换成以前缀形式表示的规则集,平均增加了近 800 条规则。

2) 分类一百万个数据包,记录 RMBRDM 算法和 PTS 算法匹配数据包所消耗的平均时间。图 2 描述了 RMBRDM 算法和 PTS 算法在匹配数据包的过程中所消耗时间的关系。在图 2 中,横坐标表示规则个数,纵坐标表示 PTS 算法平均匹配时间和 RMBRDM 算法平均匹配时间的比值。

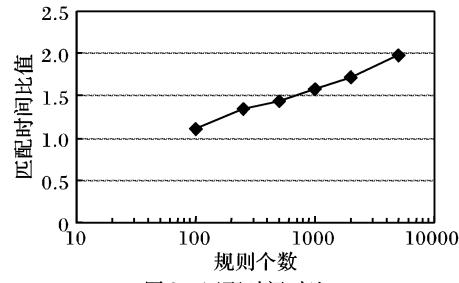


图 2 匹配时间对比

从图 2 可知,RMBRDM 算法数据包匹配时间少于 PTS 算法。当规则个数为 100 时,PTS 算法匹配时间是 RMBRDM 算法的 1.12 倍;而当规则个数增加到 1 000 时,PTS 算法匹配时间是 RMBRDM 算法的 1.59 倍。造成这种情况的主要原因是:PTS 算法构建的二叉树节点远远多于 RMBRDM 算法。这主要是由于:第一,PTS 算法构建了大量的额外的 external nodes;第二,PTS 算法将以范围形式表示的规则转换成以前缀形式表示的规则时,新增了过多规则,即增加了过多的二叉树节点。

3) 图 3 描述了 RMBRDM 算法和 PTS 算法在空间使用量的关系。在图 3 中,横坐标表示规则个数,纵坐标表示 PTS 算法空间使用量和 RMBRDM 算法空间使用量的比值。

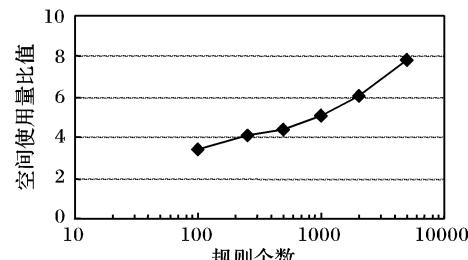


图 3 空间使用量对比

从图 3 可知,RMBRDM 算法空间使用量少于 PTS 算法。当规则个数为 100 时,PTS 算法空间使用量是 RMBRDM 算法的 3.45 倍;而当规则个数增加到 5 000 时,PTS 算法匹配时间是 RMBRDM 算法的 7.89 倍。造成这种情况的主要原因是:PTS 算法构建的二叉树节点远远多于 RMBRDM 算法。

从上述仿真实验和分析可知,RMBRDM 算法在数据结构构造、匹配效率、空间使用量等方面均优于 PTS 算法。

5 结语

针对 PTS 算法需要构建大量的 external nodes,且只支持以前缀形式表示的规则这一问题,本文提出了一种基于规则分解映射的规则匹配算法 RMBRDM。本文从标准维选取、规则分量分解映射、规则分解映射,以及二叉树构建等方面,详细地对 RMBRDM 算法进行了讨论。算法分析和仿真实验均表明,RMBRDM 算法优于 PTS 算法。

(下转第 2976 页)

秘密图像,因此,该算法具有一定的鲁棒性。对遭受攻击后恢复出的秘密图像均可以用滤波处理得到更加清晰的图像,限于篇幅不再一一列出。



图 11 剪切攻击实验



图 12 椒盐噪声攻击实验



图 13 增加对比度攻击实验

5 结语

本文给出了一种基于变参数混沌系统的图像隐藏算法,由于变参数混沌系统具有密钥数量和密钥空间大的特点,同时,能够降低混沌特性在计算机有限精度下的退化,因此该算法有效地提高了隐藏图像的安全性;此外,该算法的分区域、单一像素的隐藏方式使得融合后图像经过多种图像处理后仍能正确恢复出隐藏的秘密图像信息,证明它具有良好的鲁棒性。本文提出的隐藏技术也完全能用于数字水印的研究。

(上接第 2971 页)

参考文献:

- [1] GUPTA P, MCKEOWN N. Algorithms for packet classification [J]. IEEE Network, 2001, 15(2): 24–32.
- [2] MEINERS C R, LIU A X, TORNG E. TCAM Razor: A systematic approach towards minimizing packet classifiers in TCAMs [C]// ICNPO'07. Washington, DC: IEEE Press, 2007: 266–275.
- [3] CHE HAO, WANG ZHI-JUN, ZHENG KAI, et al. DRES: Dynamic range encoding scheme for TCAM coprocessors [J]. IEEE Transactions on Computers, 2008, 57(7): 902–915.
- [4] LU HAIBIN, SHNI S. O($\log W$) multidimensional packet classification [J]. IEEE Transactions on Networking, 2007, 15(2): 462–472.
- [5] KIM K, SHNI S. IP lookup by binary search on length [C]// IEEE International Symposium on Computer and Communication. Washington, DC: IEEE Press, 2003: 77–82.
- [6] GUPTA P, MCKEOWN N. Packet classification using hierarchical

参考文献:

- [1] YEN J, GUO J. A new chaotic key-based design for image encryption and decryption [C]// IEEE International Conference on Circuits and Systems. Washington, DC: IEEE Press, 2000: 49–52.
- [2] PAREEK N K, PATIDAR V, SUD K K. Discrete chaotic cryptography using external key [J]. Physics Letters A, 2003, 309(1/2): 75–82.
- [3] 王道顺, 齐东旭. 一种新的数字图像隐藏方案[J]. 计算机学报, 2000, 23(9): 949–952.
- [4] 张贵仓, 王让定, 章毓晋. 基于迭代混合的数字图像隐藏技术[J]. 计算机学报, 2003, 26(5): 569–574.
- [5] 赵玉霞, 康宝生. 一种基于混沌序列的数字图像隐藏算法[J]. 西北大学学报: 自然科学版, 2008, 38(2): 194–198.
- [6] 郭蔚, 蔡云飞. 一种基于融合的数字图像隐藏技术[J]. 河北工业大学学报, 2003, 32(5): 73–75.
- [7] 贺超, 赵春喜. 基于混沌的图像加密隐藏方法[J]. 长春理工大学学报: 自然科学版, 2008, 31(2): 115–117.
- [8] 李鹏, 田东平, 张楠. 基于混沌序列的数字图像隐藏技术[J]. 信息安全与通信保密, 2007(6): 222–225.
- [9] 张雪锋, 罗祖军, 周晓. 基于混沌序列的数字图像隐藏技术[J]. 西安邮电学院学报, 2006, 11(1): 75–77.
- [10] 张雪锋, 范九伦. 基于图像融合的数字图像隐藏技术[J]. 微电子学与计算机, 2007, 24(2): 188–190.
- [11] 张永红. 一种基于混沌序列的多幅图像隐藏算法[J]. 计算机工程与应用, 2008, 44(17): 182–184.
- [12] 张雪锋, 范九伦. 一种基于混沌系统的数字图像隐藏技术[J]. 计算机工程, 2007, 33(4): 134–136.
- [13] 齐东旭, 邹建成, 韩效有. 一类新的置乱变换及其在图像信息隐藏中的应用[J]. 中国科学: E 辑, 2000, 30(5): 440–447.
- [14] BENDER W, GRUHL D, MORIMOTO N, et al. Techniques for data hiding [J]. IBM System Journal, 1996, 35(3/4): 313–336.
- [15] STEFAN K, FABIEN A P. Information hiding techniques for steganography and digital watermarking [M]. London: Artech Print on Demand, 2000.
- [16] HSU C-T, WU J-L. Hiding digital watermarks in image [J]. IEEE Transactions on Image Processing, 1999, 8(1): 58–68.
- [17] PETITCOLAS F A P, ANDERSON R J, KUHN M G. Information hiding—A survey [J]. Proceedings of the IEEE, 1999, 87(7): 1062–1078.
- [18] 周志刚, 李苏贵, 刘嫣. 基于一种新的变参数混沌系统的图像加密研究[J]. 计算机应用, 2009, 29(7): 1832–1835.

- intelligent cuttings [J]. IEEE Micro, 2000, 20(1): 34–41.
- [7] SINH S, BABESCU F, VARHESE G, et al. Packet classification using multidimensional cuttings [C]// SIGCOMM'03. New York: ACM Press, 2003: 213–224.
- [8] SRINIVASAN V, VARGHESE G, SURI S, et al. Fast and scalable layer four switching [C]// SIGCOMM'98. New York: ACM Press, 1998: 191–202.
- [9] BABOESCU F, SIGH S. Packet classification for core routers: Is there an alternative to CAMs [C]// INFOCOM'03. Washington, DC: IEEE Press, 2003: 53–63.
- [10] TAYLOR D, TURNER J. Scalable packet classification using distributed crossproducting [J]. IEEE Micro, 2006, 90(5): 49–60.
- [11] PANKAJ G, NICK M. Packet classification on multiple field [J]. Computer Communication Review, 1999, 29(4): 47–60.
- [12] PAO D, LIU C. Parallel tree search: An algorithmic approach for multi-field packet classification [J]. Computer Communications, 2007, 30: 302–314.