一种IPV6环境下的高性能规则匹配算法研究

庞立会 江 峰
(国防科学技术大学计算机学院 长沙 410073)

摘 要 防火墙是确保网络安全的关键设施,而规则匹配又是防火墙的核心技术。随着网络技术的发展,互联网体系结构正逐渐从IPV4向IPV6结构发展,原有的IPV4防火墙规则匹配算法很难直接应用于IPV6网络环境,因为IPV6协议所能表示的地址范围远远超过IPV4协议对应的地址范围。因此提出了一种适用于IPV6环境的高性能规则匹配算法HiPRM(High Performance Rule Matching)。HiPRM算法的核心思想是依据规则的协议和目的端口分布特征,先把整个规则集划分成多个子规则集,再利用位选取算法对规则的源和目的IPV6地址组合的特定位进行选取,然后据此构建二叉查找规则树,最后利用规则树把多个规则子集划分成若干个更小的规则集合。而当报文匹配到某个更小的规则集合时,在小规则集中利用线性匹配法确定具体匹配的对应规则。分析和测试表明,HiPRM算法可以在时间复杂度和空间复杂度较低的情况下实现报文的高速匹配,且具有较好的规则集适应性。

关键词 防火墙,规则匹配,IPV6,二叉树
中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.03.035

Research on High Performance Rule Matching Algorithm in IPV6 Networks

PANG Li-hui JIANG Feng
(School of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract As is known to all, firewall is the core device to guarantee network security, and rule matching is one of the most important technologies to firewall. However, those high performance rule matching algorithms based on IPV4 are not suitable for IPV6 with the network derivation from IPV4 to IPV6. The main cause of this situation is that the range of IPV6 address is much bigger than the range of IPV4 address. Thus we suggested a high performance rule matching algorithm suitable for IPV6, named HiPRM (High Performance Rule Matching). HiPRM algorithm classifies the whole rule set into several parts with the protocol and destination port field of rules, and it selects special bits from the combination of source and destination IPV6 address of rules to construct binary trees. After the construction of binary trees, those rule sets are split into smaller rule sets. When a packet matches one of these smaller rule sets, the linear searching algorithm is used to find the matching rules in this small rule set. Analysis and experiment results show that HiPRM algorithm not only has good time and space performance, but also has better scalability with different rule sets.

Keywords Firewall, Rule matching, IPV6, Binary tree

1 引言

随着计算机和网络技术的不断发展和进步,国际互联网的应用与规模急速扩大,Internet当前的核心协议IPV4体系取得了巨大的成功。但IPV4协议体系也面临着IP地址即将耗尽和安全机制脆弱等不可忽视的危机和问题。为了有效解决IPV4的种种弊端,IPV6协议应运而生。IPV6作为新一代网络体系结构的核心,其采用充足的地址空间、全新的地址分配方案和安全管理机制,不仅有效地解决了IP地址资源匮乏的问题,而且较好地改进了网络的安全性和服务质量(QoS)。作为网络安全的第一道防线,防火墙在IPV4网络中得到了广泛的部署与应用,也必将在IPV6网络中成为捍卫网络安全的重要设施。

无论在IPV4环境还是IPV6环境下,防火墙等核心网络设备要正常发挥功能都无法离开规则匹配技术的支持。近年来,学术界和工业界的研发人员对规则匹配问题进行了大量研究,并针对不同环境的具体需求提出了许多有效的规则匹配算法。根据算法实现的特点,规则匹配算法大致可分为3类:1)基于特殊硬件实现的规则匹配算法,例如位向量[1](Bit Vector,BV)就是一种典型的特殊硬件实现算法,它有效地利用了硬件位层次的并行性来加速规则匹配。这类算法通常拥有TCAM等特殊硬件的支持,规则匹配速度较快,但也存在相应硬件价格昂贵、功耗较高、存储空间要求高以及难以适用于大型规则库的不足。2)基于空间分割的算法,如Hi-

到稿日期:2015-11-06 返修日期:2016-03-22 本文受国家863项目:QoS服务质量保证设计(2012AA01A50606),国家自然科学基金项目:高级持续威胁网络行为建模与检测方法研究(61303264)资助。

庞立会(1974-),男,硕士,副研究员,主要研究方向为网络安全,E-mail:lhpang@nudt.edu.cn;江 峰(1988-),男,硕士,工程师,主要研究方向为网络安全。

Cuts[2]，HyperCuts[2]和 HiPAC[3]算法等。这类算法主要依据计算几何理论和报文分类问题的几何意义来设计和实现，在面对低维和较小数目的规则匹配时具有突出的性能优势，但在规则数增大时，会造成时间和空间复杂度的急剧增大，不适合 IPV6 高维的数据包匹配[4]。3）启发式规则匹配算法，以 RFC[5]算法为代表。这类算法主要根据规则的结构和特点得出相应的一些启发式规则，并利用这些启发式规则设计相应的规则匹配算法。此类算法在时间和效率上相对比较理想，但内存消耗非常大，在 IPV6 环境中易出现存储空间爆炸现象。

本文将借鉴 IPV4 环境下 HiPAC 算法的范围分类思想，并结合 IPV6 协议本身的特点，设计一种适合 IPV6 网络环境的高性能规则匹配算法——HiPRM(High Performance Rule Matching)。

2　规则匹配问题概述

规则匹配算法作为网络设备和安全设备的核心技术，一直受到学术界和工业界的广泛关注。规则匹配的核心是报文分类问题，本质上是多域空间中的点定位问题。在研究规则匹配问题时，首先需要了解规则匹配相应问题中的 3 个基本概念：报文、搜索空间及分类规则。

报文(Packet)：网络报文通常由包含协议处理必要信息的报文头部和承载数据的报文载荷组成。报文 p 通常包含 d 个域的报文头部，报文头部的各个域分别表示为 $p[1],p[2],\cdots,p[d]$，其中每个域的取值都是特定长度的比特串，例如 128 位的 IPV6 网络层 IP 地址、16 比特的传输层端口号等。在规则匹配算法中，传统上关注的是报文头部五元组信息，包括源 IP 地址、目的 IP 地址、源端口、目的端口和协议号。

搜索空间(Search Space)：报文 p 在 d 维空间所有可能的取值构成搜索空间 S。S 的各个维度值域不同，对于 IPV6 五元组报文分类问题，$S=[0,2^{128}-1]\times[0,2^{128}-1]\times[0,2^{16}-1]\times[0,2^{16}-1]\times[0,2^{8}-1]$。

规则(Rule)：每个规则通常由 3 个部分构成，即各域的范围表示 $r[1],r[2],\cdots,r[d]$以及规则优先级 r. pri 和规则决策动作(action)r. act。

规则匹配(Rule Matching)：报文 p 与规则 r 匹配是指对于 $\forall i$，其中 $1\leqslant i\leqslant d$，有 $p[i]\in r[i]$。对于包含 n 条规则的规则集合 $R=\{r_1,r_2,\cdots,r_n\}$，p 可能与其中的多个规则匹配。

解决规则匹配问题需要设计高效的规则匹配算法，一般来说，主要从以下 3 个方面来评价一个算法的优劣。

(1)时间复杂度：规则匹配所需要的时间。时间复杂度主要反映算法的处理速度，实际应用中通常用系统吞吐量(throughput)来评价规则匹配算法的性能。单位时间内的吞吐量越高，规则匹配的性能越好。

(2)空间复杂度：算法运行所需要的存储空间大小。规则匹配算法使用的内存不仅仅指存放规则集合本身所占用的存储空间，还包括算法建立的用于查找的数据结构存储空间。考虑到实际系统的内存空间有限，规则匹配算法应尽可能压缩其数据结构以支持更多的分类规则。

(3)支持的规则规模：算法处理大规模规则集的能力。

3　HiPRM 算法的基本原理

M Kounavis 和 A Kumar 等通过实际调查，分析指出典型的企业级防火墙包含的规则数目可以达到 3000 条以上[6]。而 Gupta 和 McKeown 则进一步给出了一系列关于真实规则集特征的观测结果，这些观测结果已被广泛地引用[2]。根据 Gupta 和 McKeown 的观测结果，真实规则集一般具有下列规律[7-8]：

(1)在绝大多数规则集中，TCP 和 UDP 是最常见的协议类型，其他的协议类型有 ICMP，IGMP，(E) IGRP，GRE 和 IPINIP 等。

(2)一个包含 5000 条规则的规则集通常能够匹配一个包的规则数不超过 5 条。

(3)目的端口值大于 1023 的规则在防火墙规则集中所占的比例较小，并且分布不均匀；而目的端口值在 0～1023 之间(操作系统预定义的端口)分布比较均匀。

另外，根据 IPV6 的路由表可以得到 IPV6 的地址特征：

(1)IPV6 网络号长度分布比较集中，现有的 IPV6 地址网络号的长度大多数都是 24 位、28 位、32 位、48 位、64 位。其中，大部分网络号长度为 32 位，而网络号长度超过 64 位的地址很少。

(2)前缀为 2001 的地址占 IPV6 地址的 70%，而前缀为 2002 的地址占 IPV6 地址的 20%，其他约占 10%。

3.1　HiPRM 算法的基本思想

根据以上 IPV6 规则集和地址特征，借鉴范围匹配思想，HiPRM 算法的基本思想如下：

(1)由于防火墙规则的协议域取值一般为 TCP 和 UDP 协议，并且 ICMP 协议也很常见，因此首先利用协议号对规则进行过滤。可以把规则分成 4 类：TCP 规则集、UDP 规则集、ICMP 规则集、其他协议规则集。

(2)再进一步使用目的端口号对根据协议号划分的规则集进行细分。IPV6 防火墙规则中目的端口号取值在 1023 以上的较少，并且在 0～1023 之间是呈均匀分布的。借鉴 HiPAC 算法的范围匹配思想，根据目的端口号取值，可以进一步将防火墙规则划分成 9 类，其中把目的端口取值为 0～1023 之间的规则划分成 8 类，而把目的端口取值为 1024～65535 的规则划为最后一类。经过协议域和目的端口号的分类，可把一个规则集分成 20 个子规则集(测试表明，按照协议特点将整个规则集分成 20 个子规则集能使时间和空间复杂度达到较好的平衡)。其中 TCP 和 UDP 协议各有 9 个规则子集，由于 ICMP 和其他协议不涉及端口号，因此 ICMP 和其他协议各占一个规则子集。

(3)继续利用比特位抽取法把这 20 个规则子集各自分成一个不多于 8 条规则的规则子集。

(4)报文经过上述分类规则匹配后，将落到一个或者多个不多于 8 条规则的规则子集中，再利用线性匹配查找算法可以很容易查找到所匹配的特定规则。

HiPRM 算法结合了范围匹配算法和线性匹配查找算法的优点，能在内存消耗不是很高的情况下达到较高的吞吐率。算法的输入参数是规则集的所有规则和叶子节点所能容纳的最大规则数 N(N 为自然数)，输出结果是相应的规则二叉树

的数据结构。HiPRM 算法的基本流程如图 1 所示。

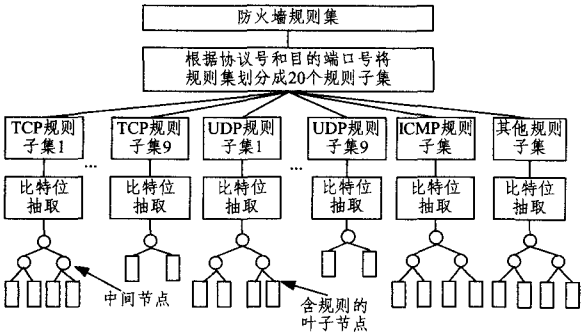

图 1 HiPRM 算法的基本流程

3.2 比特位抽取

在一条 IPV6 规则中，IPV6 报文五元组信息的长度为 296 比特，其中源 IP 和目的 IP 的地址长度分别为 128 比特，源端口号长度、目的端口号长度为 16 比特，协议号长度为 8 比特。防火墙的规则与报文的五元组信息内容密切相关，一般不同的规则具有不同的比特串，因此可以利用上述特点和比特位抽取方法对规则集实现进一步划分。

综上所述，利用协议号和目的端口号域作为首次分类标准，整个防火墙规则集将被划分为 20 个不同的规则子集，每个规则子集再使用比特位抽取法将源 IP 地址和目的 IP 地址划分成多个不大于 N 条规则的规则集合(根据规则集中通常能够匹配一个包的规则数不会超过 5 条，本文中 N 取值为 8)。

比特位抽取：由上可知，以 2001 开始的 IPV6 地址占现阶段整个 IPV6 地址的 70%，所以 IPV6 地址的前 16 位对规则的区分度不大。又由于防火墙规则中大部分 IP 地址的网络号长度为 32 位，因此把比特位的选取限定在源 IPV6 地址和目的 IPV6 地址的 16～31 位。针对这 32 个比特位，假设对 m 条规则进行位抽取，n_{i1} 表示第 $i(0<i<32)$ 个比特位为 1 的规则个数，n_{i0} 表示第 i 个比特位为 0 的规则个数。$|n_{i1}-n_{i0}|$ 表示两者之间差值的绝对值。比特位抽取的位置是 $|n_{i1}-n_{i0}|$ 取最小值时对应的 i。

被分成的 20 个子集构造树的过程如下：

(1)首先将规则源 IP 地址和目的 IP 地址域拼在一起，构成一个 256 位数(每个位有 3 种取值：0，1 和 *，其中 * 代表任意)。

(2)查找树，判断该节点是否为叶子节点，如果是，并且 N 叶子节点中规则数小于预定义的自然数 N(本文 N 值取 8)，则直接把规则放入叶子节点中，叶子节点规则数计数器增加 1，规则插入结束。

(3)如果是叶子节点，但是叶子节点中的规则数等于预定义的值 N，则需要抽取特定的比特位把叶子节点中的规则和将要插入的规则分在两个新建的叶子节点中，原叶子节点变成中间节点，成为索引叶子节点的一部分。抽取相应比特位后，按照在这个比特位的取值将规则进行进一步划分。如果规则在这个比特位的值为 1，则规则被分到左叶子节点；如果规则在这个比特位的值为 0，则其被分到右叶子结点；如果规则在这个比特位上是 *，则规则既要被分配到左叶子结点，又要被分配到右叶子节点。

(4)如果相应节点不是叶子节点，而是中间节点，则根据中间节点存储的比特位位置进行进一步操作，如果规则在这个比特位的值为 1，则继续匹配左子节点；如果这个比特位的值为 0，则继续匹配右子节点；如果规则在这个位置上是 *，则规则分两条路径，左子节点和右子节点都要匹配。转到流程(2)。

对每个子规则集中所有的规则都进行上面的操作，最后可构建出 20 棵规则二叉树。

3.3 报文匹配流程

(1)首先根据报文的协议号和目的端口号确定报文属于相应 20 个规则子集中的哪一个子集，查找这个规则子集对应的二叉树。

(2)如果节点为叶子节点，则直接线性匹配叶子节点所有规则，匹配终止。

(3)否则，若节点是中间节点，根据中间节点存储的相应比特位的位置，判断报文的源 IP 地址或者目的 IP 地址在这个位置上的值是 1 还是 0，如果是 1，则前进到左子节点；如果是 0，则前进到右子节点，转到流程(2)。

3.4 算法分析

根据前文所述，如果一个防火墙的规则数目为 n，则使用 HiPRM 算法通过协议号和目的端口号把规则集分成 20 个规则子集。每个规则子集构造一棵二叉树，算法的时间复杂度为查找二叉树所用的时间，所以 HiPRM 算法的时间复杂度为：$T_{time}=O(\log_2^{(n/20)})=O(\log_2^n)$。

HiPRM 算法的存储空间主要由规则的存储空间和规则二叉树的存储空间组成。规则的存储空间为 n，N 为每个叶子节点所容纳的规则数。假设构造的二叉树为满二叉树，二叉树节点个数为 $2^0+2^1+\cdots+2\log_2^{n/(20N)}=n/(10N)$，所以 20 棵二叉树节点个数为 $(n/(10N))*20=2n/N$。HiPRM 算法的空间复杂度为：$T_{space}=O(n+2n/N)=O(n)$。由此可以看出，该算法的空间复杂度对规则是线性的。

4 实验和结果

在目前已知的规则匹配算法中，RFC 算法是公认的规则匹配速度最快的算法，而 HyperCuts 算法是目前在时间和空间方面综合性能表现最好的算法之一。为了验证 HiPRM 算法的性能，采用网络模拟器 Network Simulator V3.22 平台对 RFC，HyperCuts 和 HiPRM 3 种算法进行了性能模拟。模拟环境所采用的操作系统为 Redhat Linux Enterprise Server 5.1，CPU 为 Intel i5 双核 2.80GHz 处理器，内存为 4GB。模拟的基本过程为随机生成不同数目的防火墙规则，然后统计当规则数不断变化时相应算法的性能参数的变化情况。实验数据为 10 次模拟情况的平均值。表 1 列出了 3 种算法下报文匹配规则的平均时间和内存平均消耗的模拟数据。

如表 1 和图 2 所示，模拟数据表明在规则匹配耗费时间方面，HiPRM 算法与 RFC 算法的数值接近，都远远优于 HyperCuts 算法。另外，HiPRM 算法的规则匹配平均时间消耗随规则数增加变化幅度不大，而 HyperCuts 算法的规则匹配平均时间却随规则数增多而剧烈增大，这表明 HyperCuts 算法难以适用于 IPV6 环境下的大规模规则集匹配，而 HiPRM 算法却可以很好地适用于类似的环境。

表 1　3 种规则匹配算法的性能比较

规则数/条	算法名称	报文匹配规则平均时间/us	内存平均消耗/MB
512	HiPRM	1.6	24.20
	RFC	1.2	27.52
	HyperCuts	12.2	40.23
1024	HiPRM	1.7	42.12
	RFC	1.4	95.17
	HyperCuts	24.2	45.17
2048	HiPRM	1.8	110.56
	RFC	2.7	403.26
	HyperCuts	47.8	55.39
4096	HiPRM	2.0	192.40
	RFC	3.9	896.85
	HyperCuts	84.9	59.62
8192	HiPRM	2.2	305.83
	RFC	6.1	2013.02
	HyperCuts	124.1	68.97
10240	HiPRM	2.3	420.75
	RFC	8.5	3276.39
	HyperCuts	317.8	83.02

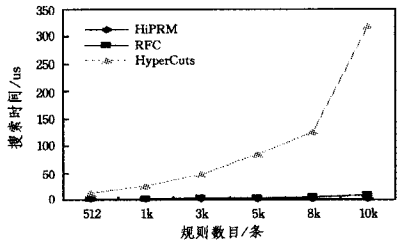

图 2　3 种算法的数据包匹配规则的平均时间消耗对比

在算法的内存空间消耗方面，HiPRM 算法所消耗的内存空间远小于 RFC 算法所消耗的，而且 RFC 算法所消耗的内存空间会随着规则数的增加而急剧增大，因此其难以适应 IPV6 环境下的大规模规则集匹配，如图 3 所示。

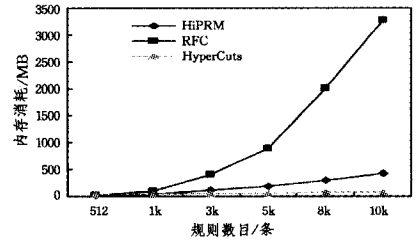

图 3　3 种算法平均消耗的内存空间对比

为了验证 HiPRM 规则匹配算法对大规则集的支持能力，利用一台 Linux 服务器构建了基于 HiPRM 算法的 IPV6 防火墙原型系统（简称 HiPRM 防火墙），其硬件配置如下：CPU 为 Intel i5 双核 2.80GHz 处理器，内存为 4GB，采用 Broadcom NetXtreme 千兆自适应网卡。操作系统为 Redhat Linux Enterprise Server5.1，内核版本为 2.6.32。

在执行规则匹配算法测试之前，需要构造用于实际测试的规则集。测试规则集主要依据文献[9]中所提出的方法进行构造。ClassBench 由华盛顿大学路易斯分校的 Taylor 等人设计开发，其产生的数据集在规则匹配研究中被广泛使用，是目前较为权威的规则库的产生和测试平台。ClassBench 主要通过对路由器、防火墙规则集等进行分析，用统计的方法得到了源端口号、目的端口号、协议域、源 IP 地址、目的 IP 地址的分布规律，并利用相应规律构造类似于现实应用中的模拟测试规则集。鉴于该文献中所提出的方法主要是针对 IPV4 地址类型的，依据前面所提到的 IPV6 地址分布特点，使用 IPV6 骨干路由表中的 IPV6 地址对其中的 IPV4 地址进行替换，以确保生成的规则集依然具有一定的代表性。

针对规则匹配问题，虽然近年来学术界提出了很多匹配算法，但是在实践中由于诸多原因，这些算法很少被防火墙等实际部署的核心网络安全设备应用。目前被防火墙等核心网络安全设备所采用的规则匹配算法依然是最简单的顺序查找算法[10]，例如 Linux 操作系统的 netfilter/iptables 防火墙，FreeBSD 操作系统的 ipfw 防火墙。本文选择 Linux 操作系统自带的 Netfilter/IP6tables 防火墙作为实际测试的比较对象。

测试方法：利用 iperf 工具生成相应的测试报文，并使用 ClassBench 方法构造的规则测试集，分别对 Linux 操作系统自带的 Netfilter/IP6tables 防火墙和 HiPRM 防火墙进行各种规模下的规则集吞吐量测试，并分析比较了两种 IPV6 防火墙的性能。测试结果如表 2 所列。

表 2　两种防火墙吞吐量的测试结果/Mb/s

吞吐量 \ 规则数 / 算法	32	128	512	1024	4096	8192	10240
IP6tables	932	866	722	613	395	115	42
HiPRM	930	902	862	857	848	840	824

由表 2 可知，当规则数较小时，IP6tables 和 HiPRM 防火墙的性能相当，但随着规则数的逐渐增多，IP6tables 防火墙的性能急剧下降，而 HiPRM 防火墙的性能下降到一定程度后趋于平稳，如图 4 所示。由此可知，Linux 系统的 IP6tables 防火墙在处理小规则集时非常高效，但是当面对大规则集时，它的效率明显下降；而使用 HiPRM 算法的防火墙却不存在类似问题，其完全可满足一般企业级 IPV6 防火墙的需求。

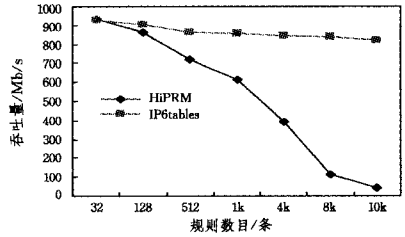

图 4　HiPRM 与 IP6tables 防火墙吞吐量的比较

结束语　针对原有的 IPV4 防火墙规则匹配算法很难直接适用于 IPV6 网络环境的现状，提出了一种适用于 IPV6 网络环境的高性能防火墙规则匹配算法——HiPRM。HiPRM 算法结合统计分类、范围匹配查找和线性查找的优势，不仅在时间复杂度和空间复杂度很低的情况下实现了对防火墙规则的高速匹配，而且可以很好地适应规则集的规模变化。本文的后续工作是探索融合相应的智能算法，进一步提高性能。

参 考 文 献

[1] LAKSHMAN T V, STILIADIS D. High Speed Policy Based Packet Forwarding Using Efficient Multidimensional Range Matching [J]. ACM Computer Communication Review, 1998, 28(4): 203-214.

[2] GUPTA P, MCKEOWN N. Packet Classification on Multiple Fields [J]. ACM SIGCOMM Computer Communication Review,

1999,29(14):147-160.

[3] HiPAC [EB/OL]. (2005-11-8) [2014-12-5]. http://www.hipac.org.

[4] WANG Z L,WU Z J,YI L. HigH-Dimension Large-scale Packet Matching Algorithm in IPV6 [J]. ACTA Electronica Sinica, 2013,41(11):2181-2186. (in Chinese)
王则林,吴志健,尹兰. IPV6 环境下的高维大规模分类算法[J]. 电子学报,2013,41(11):2181-2186.

[5] SINGH S,BABOESCU F,VARGHESE G,et al. Packet Classification Using Multi-dimensional Cutting[C]// Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, 2003. Karlsruhe:ACMSIGCOMM,2003:213-224.

[6] BABOESCU F,SINGH S,VARGHESE G. Packet Classification for Core Routers:Is There an Alternative to CAMs? [C]//Proc of IEEE Infocom, 2003. San Francisco, IEEE Infocom, 2003: 53-63.

[7] HAN W T,YI P,ZHANG X. Hybrid Cutting Algorithm for Packet Classification [J]. Journal of Software, 2014, 25(11): 2616-2626. (in Chinese)
韩伟涛,伊鹏,张霞. 一种采用混合切分法的报文分类算法[J]. 软件学报,2014,25(11):2616-2626.

[8] KOGAN K,NIKOLENKO S,ROTTENSTEICH O,et al. SAX-PAC(Scalable and Expressive Packet Classification)[C]// Proceedings of the 2014 ACM Conference on SIGCOMM, 2014. Chicago,IL,USA:ACM Press,2014:15-26.

[9] TAYLOR D E,TURNER J S. ClassBench:A Packet Classification Benchmark [J]. IEEE/ACM Trans. on Networking, 2007, 15(3):499-511.

[10] HAGER S,SELENT S,Scheuermann B. Trees in the List:Accelerating List-based Packet Classification Through Controlled Rule Set[C]//Proc of ACM CoNEXT,2014. Sydney,Australia: ACM Press,2014:101-107.

(上接第 139 页)

的计算方法和三维 MUSIC 估算函数更简单,算法复杂度更低。

表 2 占用 CPU 时间对比

算法	文献[6]的方法	三维 MUSIC 方法	CRLB 方法	本文方法
时间	2.7752E-4	24.7183	2.3087E-4	1.6764E-4

结束语 本文提出一种新的闭环式算法用于估算基于均匀圆阵列的单基站三维测向。与以往测向估算的算法不同,所提出的算法适用于任意数量的传感器,且无需使用均匀圆阵列的中心对称性就可以估算所有的三维测向参数,计算方法简单。相比多种类似算法,所提算法的性能可与三维 MUSIC 算法的性能相媲美,且其计算复杂度更低,实现速度更快。

参 考 文 献

[1] JI Y. Research on the Direction Finding Technology of Uniform Circular Array[D]. Xi'an:Xidian University,2009. (in Chinese)
季宇. 均匀圆阵列测向技术研究[D]. 西安:西安电子科技大学,2009.

[2] TIAN X H,LIAO G S,WANG H Y,et al. Locating and Tracking of 3D Moving Targets Using Single Base Station[J]. Journal of Electronics & Information Technlogy,2004,26(9):1364-1370. (in Chinese)
田孝华,廖桂生,王洪洋. 利用单基站对三维运动目标测向与跟踪[J]. 电子与信息学报,2004,26(9):1364-1370.

[3] WU Y,SO H C. Simple and Accurate Two-Dimensional Angle Estimation for a Single Source With Uniform Circular Array [J]. IEEE Antennas & Wireless Propagation Letters,2008,7: 78-80.

[4] LIAO B,WU Y T,CHAN S C. A Generalized Algorithm for Fast Two-Dimensional Angle Estimation of a Single Source With Uniform Circular Arrays[J]. IEEE Antennas & Wireless Propagation Letters,2012,11(2):984-986.

[5] EUN-HYON B,KYUN-KYUNG L. Closed-Form 3D Localization for Single Source in Uniform Circular Array with a Center Sensor[J]. IEICE Transactions on Communications,2009,92-B (3):1053-1056.

[6] WU Y,WANG H,HUANG L,et al. Fast algorithm for three-dimensional single near-field source localization with uniform circular array[C]//2011 IEEE CIE International Conference on Radar (Radar). IEEE,2011:350-352.

[7] WU Y,WANG H,HUANG L,et al. Fast algorithm for three-dimensional single near-field source localization with uniform circular array[C]//2011 IEEE CIE International Conference on Radar (Radar). IEEE,2011:350-352.

[8] LI Y,XING X F,HUANG X T,et al. Design of Simulation System for Uniform Circular Array [J]. Computer Simulation, 2005,22(12):238-241. (in Chinese)
李颖,邢向飞,黄晓涛,等. 均匀圆阵列天线仿真系统设计[J]. 计算机仿真,2005,22(12):238-241.

[9] DING W D,MA Y L. Using MUSIC Algorithm to Estimate the DOA of Coherent Source Via Virtual Array Transformation[J]. Journal of Microwaves, 2008, 24(2):27-30. (in Chinese)
丁卫安,马远良. 虚拟阵列变换法解相干信号 MUSIC 算法研究[J]. 微波学报,2008,24(2):27-30.

[10] DAN Z L,LIU F W. Study for the Localization Algorithms Based on the Antenna Array in Wireless Sensor Networks[J]. Journal of Chinese Computer Systems,2014,35(1):85-88. (in Chinese)
单志龙,刘方伟. 天线阵列用于无线传感器网络的节点测向算法[J]. 小型微型计算机系统,2014,35(1):85-88.

[11] YANG H L. Research on Direction Finding Algorithm Based on Uniform Circular Array[D]. Xi'an:Xidian University,2014. (in Chinese)
杨洪亮. 基于均匀圆阵的测向算法研究[D]. 西安:西安电子科技大学,2014.

[12] XIA T,XIE J H. On the pattern of circular antenna array configuration in electronic reconnaissance[J]. Journal of Chongqing University of Posts and Telecommunications(Natural Science Edition),2011,23(3):310-314. (in Chinese)
夏添,谢俊好. 电子侦察中圆形天线阵列配置方式仿真分析[J]. 重邮电大学学报(自然科学版),2011,23(3):310-314.