# 未知单协议数据帧的地址分析与研究

郑 杰[1] 朱 强[2]
(电子科技大学计算机科学与工程学院 成都 611731)[1] (重庆电子工程职业学院 重庆 401331)[2]

**摘 要** 网络协议是网络通信中一系列标准的集合,未知协议的识别和分析对网络监管、保障网络安全具有重大意义。协议识别技术多种多样,但在协议的分析识别过程中,为了实现协议的简单高效识别,通常需要将未知混合多协议分离为单协议,然后再进行进一步的识别。在将未知混合数据帧分离为单协议的基础上,提出了一种高效的确定单协议位置信息的方法,即进一步将单协议的数据帧按地址分为点对点数据帧,从而实现未知协议的最终识别。最后通过分析ARP、TCP数据对该方法进行评估,结果表明采用该方法可以找到2/3以上的地址信息。

**关键词** 协议识别,协议分离,单协议,数据帧,地址信息
**中图法分类号** TP391.4 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.11.038

## Analysis and Research on Address Message of Unknown Single Protocol Data Frame

ZHENG Jie[1] ZHU Qiang[2]
(Department of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)[1]
(Chongqing College Academy of Electronic Engineering, Chongqing 401331, China)[2]

**Abstract** Network protocols are sets of standards for certain network communications. The protocol identification and analysis have great significance for network management and security. The technologies of protocol identification are varied, but in the process of protocol identification, in order to simplify the identification process and improve the efficiency of protocol identification, it usually needs to separate the unknown mixed multi-protocol into single protocol, and then makes further identification. This paper presented an efficient method to determine the single protocol address message based on the previous work to separate unknown mixed data frame into single protocol. By this way the data frames of single protocol are split into point to point data frame according to the address, and then the final identification of unknown protocol is achieved. In the end, we evaluated the method by analyzing the ARP and TCP data. The results show that this method can find out more than 2/3 address information.

**Keywords** Protocol identification, Separate protocol, Single protocol, Data frame, Address message

## 1 引言

无线网络已成为当前重要的一种网络接入方式。无线网络的发展和无线接入技术的广泛应用,加之无线网络的移动性和灵活性,使无线网络用户成倍增长[1]。无线网络协议的脆弱性和通信的隐蔽性使无线网络存在隐患,随着无线网络技术的快速发展和广泛应用,只有增强对无线网络的管理以及不断对其优化才能使无线网络健康、快速地继续发展下去。而未知协议发现技术可以在某一无线环境下发现存在的未知协议,为无线网络的管理控制提供有效的技术和数据支持,限制当前无线网络中过多的专用协议,优化无线网络环境。

在协议识别实现的早期,协议识别绝大部分是按照端口映射的方式实现的[2]。这样做的依据是大多数操作系统和应用软件都是在假定RFC被严格遵守的情况下编写的。在协议规范公开的同时已经设定好该协议默认使用的通信端口,并且坚定使用者们都会遵守这些规范,如HTTP使用80端口、FTP使用21端口等。因此最早的协议识别技术是基于端口的[3],该技术能识别在IANA(Internet Assigned Numbers Authority)[4]中注册端口号的协议。大量协议采用自由端口,但是随着网络协议的发展,一些新的协议表现出了新的特点,主要有:(1)不使用固定端口进行通信;(2)复用公开端口进行私有协议通信;(3)采用已知公开协议的传输工具。这些新的特点使得原有的基于端口映射的协议识别技术的局限性越来越明显,导致基于端口的协议识别技术已经不适用于现在的网络环境。文献[5-7]详细论述了导致基于端口算法失效的原因,正是由于这些原因,基于端口识别协议的准确性已经低于50%。

深度包检测技术(Deep Packet Inspection, DPI)[8,9]最主要的特点[10]就是对网络报文的数据包进行更深层次的扫描和匹配。该技术的实现原理与普通基于特征字的检测方法基本一致,只是其相比普通检测技术检测得更加深入,识别率也自然得到了提高。此外,深度包检测技术灵活地应用了其他

到稿日期:2014-11-20 返修日期:2014-12-23 本文受中国工程物理研究院科技发展基金(2012A0403021),NSAF联合基金(U1230106),国家信息安全发展计划(2013F098)资助。

郑 杰(1976-),男,博士生,主要研究方向为信号处理、信息分析,E-mail:1120103@qq.com;朱 强 讲师,主要研究方向为计算机应用、信息系统。

检测方法的优势，通常 DPI 检测系统会进行较简单的端口识别，当端口识别无法识别出来时再对报文进行基于特征字技术的识别。另外，DPI 技术还融入了应用层网关检测技术，用于解决控制流与数据流分离的 P2P 问题。

Subhabrata Sen 等人[9]通过分析 5 种 P2P 协议的相关文档和实际报文流量，确定协议的应用层特征串，该方法可以较准确地识别出应用层协议的特征串，但是只适用于有公开协议规则的协议。王一鹏[11]等人提出了一种基于语义的协议识别方法，这种方法通过 3-gram 对原始数据集进行切分，通过语义进行分词，然后利用 LDA 算法完成格式无关单元的过滤和协议特征关键字的生成，最后完成聚类和协议格式的提取。这种方法计算过程过于复杂，花费的时间代价过大，并且从语义层面实现关键字的提取，更适合于文本协议或者语义已知的二进制协议。王勇[12]等人提出一种基于关键字的提取方法，这种方法既适用于文本协议也适用于二进制协议；对于二进制单协议的数据输入，将其按字节进行切分和拼接得到特定协议的关键字，但是这种方法在实际的操作过程中对拼接成的长串进行切分时很难把握切分的原则。

从理论上说，基于负载的协议识别方法可以通过分析协议规范和实际交互的报文得到协议的特征，并且该方法的准确性是目前所有算法中最高的，文献[12-14]所研究的算法的误报率均小于 10%，但该类算法的时空复杂度也是目前所有算法中最高的，并且随着待识别协议数量的增长而增长。使用基于负载的算法需要不断地跟踪待识别协议的发展，更新的工作量非常大。因此，该类算法通常被用于待识别数量较少的协议，且工作量较大。

在一般情况下（加密技术流量除外），使用深度包检测技术能够使识别率提高到 95%。深度包检测技术最主要的优势就是能够准确地识别出使用不定端口的 P2P 流量（当前 P2P 应用正在向这个趋势发展），而且识别的准确率比较高。但是，深度包检测技术主要是针对现有 P2P 的应用，针对版本更新或新加入的协议，识别效果就会下降，只能通过删改旧规则、增加新规则来弥补，因此其扩展性不是很好。另外深度包解析技术深层挖掘网络应用层数据，有可能侵犯到用户的隐私，很多用户会因为隐私问题抵触这种技术。并且面对网络加密数据时，采用基于特征字的识别方式很难达到识别的目的。深度包检测技术的核心技术也是基于特征字的检测技术，所以其对于加密数据也很难达到识别的目的。

1975 年 Aho 和 Corasick 提出基于确定性有限自动机（DFA）理论的模式匹配算法[15]，这一算法产生于贝尔实验室，是模式匹配问题中一种经典的算法。该算法应用有限自动机巧妙地将字符比较转换为状态转移。模式匹配算法是一种基础的算法，它包括单模式匹配算法[16]和多模式匹配算法[17]。单模式匹配算法是在字符串中寻找某个特定模式的过程。如果要匹配多个模式，那么有几个模式就需要遍历几次。而多模式匹配算法经过一次遍历就可以对多个模式进行匹配，从而大大提高了匹配效率。当然多模式匹配算法也适用于单模式的情况。随着网络协议的复杂化，一条特征可能匹配或部分匹配很多条规则，如果采用单模式匹配，在匹配每条规则时都需要重新运行匹配算法，效率很低。因此，多模式匹配算法在协议识别的应用[18]中有很大的优势。

在无线网络环境中，无线网络是采用类似广播性质实现信号传输，使得无线网络数据帧更为复杂。同时，随着无线网络通信技术的发展，无线通信的标准也变得繁杂多样，并且很多标准之间不兼容，这也对无线网络的协议的识别带来更多的困难。

本文基于无线网络环境这一热点，并根据前期利用数据链路层比特流数据的关联特征，提出了无线协议发现识别方法，在将未知混合多协议分离为单协议的基础上[19]，进一步分析，找到协议的地址信息，最后将单协议的数据帧按地址分为点对点数据帧，从而为无线网络安全监管、网络服务质量（QoS）保障等提供技术支撑。

## 2 准备工作

### 2.1 名词解释

MAC（Medium/Media Access Control）地址：具有全球唯一性的标识网卡的一串符号，一般译为网卡的物理地址，用来表示互联网上每一个站点的标识符，采用十六进制数表示，共 6 个字节（48 位）。

IP 地址：每个连接在 Internet 上的主机分配的一个 32bit 地址。按照 TCP/IP 协议规定，IP 地址用二进制来表示，每个 IP 地址长 32bit，比特换算成字节，即 4 个字节。

协议指纹：用于唯一标示一个协议的序列，是协议数据报文特征的具体化表达。该序列出现在协议的每一条消息报文中，可用于在数据流中标示该协议。

局域网（Local Area Network，LAN）：是指在某一区域内由多台计算机互联成的计算机组，可以实现文件管理、应用软件共享、打印机共享、工作组内的日程安排、电子邮件和传真通信服务等功能。

WEKA（Waikato Environment for Knowledge Analysis）：是一款免费的、非商业化的（与之对应的是 SPSS 公司的商业数据挖掘产品 Clementine）的、基于 JAVA 环境下开源的机器学习（machine learning）以及数据挖掘（data minining）软件。

DARPA（Defense Advanced Research Projects Agency）：是美国国防部重大科技攻关项目的组织、协调、管理机构和军用高技术预研工作的技术管理部门，主要负责高新技术的研究、开发和应用。

### 2.2 寻找单协议数据帧地址的原理

用聚类算法（k-means）将混合未知多协议数据帧分为单协议，并用评估算法确定所得到的类簇是比较可信的单协议数据帧，为进一步分析，需要将单协议的数据帧按地址分为点对点数据帧，而其关键是找到协议的地址信息。

假定算法的输入协议数据帧具有如下特性：

（1）数据帧中存在源地址信息和目的地址信息，可以是物理地址或网络地址；

（2）地址信息所占的字节数未知，假设不少于 2 个字节（不排除协议只用 1 个字节作为地址的可能性）。

在单协议情况下，假设有 $n$（足够大）条该协议的数据帧。

（1）寻找数据帧中的这些列：其中出现字符的种类数大于 1 小于 $K$（默认值 256），$K$ 作为可变参数。

（2）假设得到 $S$ 列，循环处理每一列，挑选出这样的列对加到集合 $R$ 中：

在 $S_i$ 列中，有超过 $w\%$（默认 60%）的字符在 $S_j$ 列中的不同位置也出现了，并且在 $S_j$ 列中，有超过 $w\%$（默认 60%）的

字符在 $S_i$ 列中的不同位置也出现了，则将 $S_i$，$S_j$ 加入集合 $R$。

(3)集合 $R$ 中得到的列就是地址列的候选集。

(4)如果集合 $R$ 中不止 2 列，则相邻的列进行拼接。

(5)将 $w$ 值设为从 10 到 90，分别计算出相应的地址对。

(6)对比分析得到的地址对，找出最优解。

可设置的参数：$K$，$w$。

## 3 求解协议帧位置信息的实现

(1)数据输入：将切分好帧的二进制数据帧转换为对应的十六进制格式，以 2 个字节作为处理单元，构成一个具有 $n$ 行、$m$ 列的二维矩阵，每个元素就是 2 个字节所对应的十六进制字符，用字符串表示。

(2)定义最小处理单元对象：TwoByte，属性有：

```
Class TwoByte {
    /**此字节所在行*/
    public int row = 0;
    /**此字节所在列*/
    public int line = 0;
    /**此字节在该列中出现的频率*/
    public float frequence = 0f;
    /**此字节的内容*/
    public String twoByte="";
    /**在该列中，出现此字节的行的编号集合*/
    public HashSet<Integer> alist = new HashSet<Integer>();
}
```

(3)建立 TwoByte 的 $n$ 行、$m$ 列的二维数组，将输入的数据帧的每 2 个字节的内容赋给 TwoByte 对象的 twoByte 域，并且记录该字符串所在的行和列。

(4)循环遍历 TwoByte 二维数组，按列统计，统计每一列中每个字符串出现的次数以及哪些行出现过该字符串。将出现的次数赋值给 TwoByte 的 num 域，将出现过该字符串的行加入到 TwoByte 的 alist 集合中。这样就得到了每个字符串在那一列中出现的次数以及出现过该字节的数据帧的行号。

(5)设定阈值 min_numOfperLine(默认 1)和 max_numOfperLine(默认 256)，筛选出字符串种类数大于 min_numOfperLine 且小于 max_numOfperLine 的列作为下一步的输入。

(6)假设经以上步骤得到 $S$ 列，循环处理每一列，设定阈值 $w\%$(默认 60%)以及结果集 $R$，挑选出这样的列对加到集合 $R$ 中：

在 $S_i$ 列中，有超过 $w\%$的字符也出现在 $S_j$ 列中的不同位置，并且在 $S_j$ 列中，有超过 $w\%$的字符也出现在 $S_i$ 列中的不同位置，则将 $S_i$，$S_j$ 加入集合 $R$。

(7)集合 $R$ 中得到的地址对即为要求的候选地址所在的列。如果集合 $R$ 中不止 2 列，则相邻的列进行拼接。

(8)为更准确地找到地址所在的位置，将 $w$ 的值设为 50 到 95，对比分析 $R$ 中的地址对，找出最优解。

## 4 实验结果与分析

### 4.1 实验条件

本文针对双方简单通信的场景，提出了一种基于特征串的单协议分类方法。其中数据帧的 $n$-gram 切分、拼接以及特征选择算法都是通过 JAVA 代码实现的，聚类算法是通过 WEKA 中自带的 k-means 算法实现的，运行环境为 Win7 操作系统，Pentium(R) Dual-Core CPU E5800，2GB 内存。

### 4.2 实验数据集

本次实验中所使用的数据集是来自林肯实验室的 DARPA 数据集。数据集分为 inside. tcmpdump 和 outside. tcpdump 两部分。本文选用 inside. tcmpdump 作为实验数据集，其中共包含 1130829 条数据帧。为验证本算法的有效性，实验使用了 2000 条 ARP 数据帧和 10000 条 TCP 数据帧分别进行了验证，以下是实验结果。

### 4.3 实验结果

#### 4.3.1 2000 条 ARP 数据帧地址位置确定实验

数据输入：2000 条 ARP 数据帧，取前 42 字节(数据帧最短为 42 字节)，2 字节作为最小处理单元，一共有 21 列。

min_numOfperLine=1，max_numOfperLine=256，$w$ 从 50 到 90 的实验结果如表 1 所列(列号从 0 开始)。

表 1 从 ARP 数据帧中提取的地址对

| w 值 | 候选地址列 | 地址对 | 拼接地址对 |
|---|---|---|---|
| 50 | [1,2,3,4,5,11,12,13,14,17,18,19] | (1,4)(1,12)(1,17)<br>(2,5)(2,13)(2,18)<br>(3,11)<br>(4,1)(4,12)(4,17)<br>(5,2)(5,13)(5,18)<br>(11,3)<br>(12,1)(12,4)(12,17)<br>(13,2)(13,5)(13,18)<br>(14,19)<br>(17,1)(17,4)(17,12)<br>(19,14)<br>(18,2)(18,5)(18,13) | [1 2,4 5];[1 2,12 13];[1 2,17 18]<br>[4 5,12 13];<br>[4 5,17 18]<br>[12 13,17 18]<br>[17 18 19,12 13 14] |
| 60 | [1,2,4,5,12,13,14,17,18,19] | (1,4)(1,12)(1,17)<br>(2,5)(2,13)(2,18)<br>(4,1)(4,12)(4,17)<br>(5,2)(5,13)(5,18)<br>(12,1)(12,4)(12,17)<br>(13,2)(13,5)(13,18)<br>(14,19)<br>(17,1)(17,4)(17,12)<br>(19,14)<br>(18,2)(18,5)(18,13) | [1 2,4 5];[1 2,12 13];[1 2,17 18]<br>[4 5,12 13];<br>[4 5,17 18]<br>[12 13,17 18]<br>[17 18 19,12 13 14] |
| 70 | [1,2,4,5,12,13,14,17,18,19] | (1,4)(1,12)(1,17)<br>(2,5)(2,13)(2,18)<br>(4,1)(4,12)(4,17)<br>(5,2)(5,13)(5,18)<br>(12,1)(12,4)(12,17)<br>(13,2)(13,5)(13,18)<br>(14,19)<br>(17,1)(17,4)(17,12)<br>(19,14)<br>(18,2)(18,5)(18,13) | [1 2,4 5];[1 2,12 13];[1 2,17 18]<br>[4 5,12 13];<br>[4 5,17 18]<br>[12 13,17 18]<br>[17 18 19,12 13 14] |
| 80 | [1,2,4,5,12,13,14,17,18,19] | (1,4)(1,12)(1,17)<br>(2,5)(2,13)(2,18)<br>(4,1)(4,12)(4,17)<br>(5,2)(5,13)(5,18)<br>(12,1)(12,4)(12,17)<br>(13,2)(13,5)(13,18)<br>(14,19)<br>(17,1)(17,4)(17,12)<br>(19,14)<br>(18,2)(18,5)(18,13) | [1 2,4 5];[1 2,12 13];[1 2,17 18]<br>[4 5,12 13];<br>[4 5,17 18]<br>[12 13,17 18]<br>[17 18 19,12 13 14] |
| 90 | [1,2,4,5,12,13,14,19] | (1,4)(1,12)<br>(2,5)(2,13)<br>(4,1)(4,12)<br>(5,2)(5,13)<br>(12,1)(12,4)<br>(13,2)(13,5)<br>(14,19)<br>(19,14) | [1 2,4 5];[1 2,12 13]<br>[4 5,12 13] |

从表 1 的拼接地址对可以看出，列号从 0 开始，程序中的

1 2,4 5,12 13,17 18 为地址列。对应于输入数据的列 2 3 4 5,8 9 10 11,24 25 26 27,34 35 36 37 为地址列。

根据表 1 的结果可以看出，采用本文的寻找地址算法找到的 ARP 数据帧的地址列有：2 3 4 5,8 9 10 11,24 25 26 27,34 35 36 37。图 1 给出了 2 条 ARP 数据帧，根据 ARP 数据帧的格式容易知道，0 1 2 3 4 5 列是目的 MAC 地址，6 7 8 9 10 11 列是源 MAC 地址，22 23 24 25 26 27 为源 MAC 地址列，28 29 30 31 为发送方 IP 地址列，32 33 34 35 36 37 为目的 MAC 地址列，38 39 40 41 为接收方 IP 地址列。分析 ARP 数据帧结构，将该方法找到的地址列与 ARP 数据帧真实地址列相应地列于表 2 中，通过对比来评价该寻址方法的性能。

目的MAC地址　　源MAC地址

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ff | ff | ff | ff | ff | ff | 00 | 10 | 5a | 9c | b2 | 54 | 08 | 06 | 00 | 01 | 08 | 00 | 06 | 04 | 00 |
| 00 | 10 | 5a | 9c | b2 | 54 | 00 | c0 | 4f | a3 | 57 | db | 08 | 06 | 00 | 01 | 08 | 00 | 06 | 04 | 00 |

源MAC地址列　　目的MAC地址列

| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 01 | 00 | 10 | 5a | 9c | b2 | 54 | ac | 10 | 70 | 64 | 00 | 00 | 00 | 00 | 00 | 00 | ac | 10 | 70 | 14 |
| 02 | 00 | c0 | 4f | a3 | 57 | db | ac | 10 | 70 | 14 | 00 | 10 | 5a | 9c | b2 | 54 | ac | 10 | 70 | 64 |

图 1　2 条 ARP 数据帧的地址信息

表 2　ARP 识别数据与真实数据地址对比

| | 目的 MAC 地址 | 源 MAC 地址 | 源 MAC 地址列 | 目的 MAC 地址列 |
|---|---|---|---|---|
| 识别数据 | 2 3 4 5 | 8 9 10 11 | 24 25 26 27 | 34 35 36 37 |
| 真实数据 | 0 1 2 3 4 5 | 6 7 8 9 10 11 | 22 23 24 25 26 27 | 32 33 34 35 36 37 |

根据表 2 的数据对比可知，采用该算法虽然没有把所有的地址列都找出来，但是对于所找出的每一个地址段，该方法都可以找出该地址列的 2/3 的列，对这些地址列信息的处理，足以作为将这些数据帧分离为点对点数据的依据。

4.3.2　10000 条 TCP 数据帧地址位置确定实验

数据输入：10000 条 TCP 数据帧，取前 60 字节(数据帧最短为 60 字节)，2 字节作为最小处理单元，一共有 30 列。

min_numOfperLine＝1，max_numOfperLine＝256，$w$ 从 50 到 90 的实验结果如表 3 所列。

表 3　从 TCP 数据帧中提取的地址对

| w | 候选地址列 | 地址对 | 拼接地址对 |
|---|---|---|---|
| 50 | [0,1,2,3,4,5,13,14,15,16] | (0,3)(1,4)(2,5)(3,0)(4,1)(5,2)(13,15)(14,16)(15,13)(16,14) | [0 1 2,3 4 5][13 14,15 16] |
| 60 | [0,1,2,3,4,5,13,14,15,16] | (0,3)(1,4)(2,5)(3,0)(4,1)(5,2)(13,15)(14,16)(15,13)(16,14) | [0 1 2,3 4 5][13 14,15 16] |
| 70 | [0,1,2,3,4,5,13,14,15,16] | (0,3)(1,4)(2,5)(3,0)(4,1)(5,2)(13,15)(14,16)(15,13)(16,14) | [0 1 2,3 4 5][13 14,15 16] |
| 80 | [1,2,4,5,13,14,15,16] | (1,4)(2,5)(4,1)(5,2)(13,15)(14,16)(15,13)(16,14) | [1 2,4 5][13 14,15 16] |
| 90 | [13,14,15,16] | (13,15)(14,16)(15,13)(16,14) | [13 14,15 16] |

从表 3 的拼接地址对可以看出，程序中的 0 1 2,3 4 5,13 14,15 16 为地址列。对应于输入数据的列为：0 1 2 3 4 5,6 7 8 9 10 11,26 27 28 29,38 39 40 41。图 2 列出了 2 条 TCP 数据帧，根据 TCP 数据帧的格式容易知道，0 1 2 3 4 5 列是目的 MAC 地址，6 7 8 9 10 11 列是源 MAC 地址，26 27 28 29 为发送方 IP 地址列，38 39 40 41 为接收方 IP 地址列。

目的MAC地址　　源MAC地址

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | 10 | 7b | 38 | 46 | 33 | 00 | 10 | 5a | 9c | b2 | 54 | 08 | 00 | 45 | 00 | 00 | 2c | 7c | 00 | 40 |
| 00 | 10 | 5a | 9c | b2 | 54 | 00 | 10 | 7b | 38 | 46 | 33 | 08 | 00 | 45 | 00 | 00 | 2c | 4b | 0a | 00 |

发送方IP　　接收方IP

| 21 | ... | 26 | 27 | 28 | 29 | ... | 38 | 39 | 40 | 41 | ... | 59 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | ... | ac | 10 | 70 | 64 | ... | 00 | 05 | 00 | 94 | ... | b4 |
| 00 | ... | ce | fb | 12 | 37 | ... | 46 | 74 | b0 | bf | ... | 00 |

图 2　2 条 TCP 数据帧地址信息

对 TCP 数据帧的识别结果展示于表 4 中，通过对比表 4 中 TCP 数据帧的识别数据与 TCP 数据帧的真实数据的地址列可知，采用本算法找出的地址列正好全部是 TCP 数据帧的地址列，因此采用本方法对 TCP 数据帧的地址列的识别达到 100%。通过进一步分析，这些列可以作为将数据帧分离为点对点数据的依据。

与传统的采用位置差关联规则[20]来寻找数据帧位置信息的方法相比，采用本方法可以有效地简化地址信息寻找流程，减少了频繁计数和判断过程，大大节省了寻找地址信息所需要的时间。同时，传统位置差关联规则寻找地址信息不能直接得到地址列，而是通过支持度参数曲线来判断，这给地址信息的确认带来一定的困难，而采用本方法可以直接得到数据帧的地址列，可以直接通过同数据帧的真实地址的信息进行比较，直观地判断该方法寻找地址信息的有效性。

表 4　TCP 数据帧识别数据与真实数据地址对比

| 地址名称 | 识别数据列 | 真实数据列 |
|---|---|---|
| 目的 MAC 地址 | 0 1 2 3 4 5 | 0 1 2 3 4 5 |
| 源 MAC 地址 | 6 7 8 9 10 11 | 6 7 8 9 10 11 |
| 发送方 IP | 26 27 28 29 | 26 27 28 29 |
| 接收方 IP | 8 39 40 41 | 8 39 40 41 |

**结束语**　本文在将未知混合多协议分离为单协议的基础上，提出了一种寻找未知协议地址信息的方法，采用该方法可以有效地找到未知单协议的地址信息，从而进一步实现将未知单协议分离为点对点数据，实现对混合未知多协议的最终识别。采用该方法可以对 ARP 数据帧的地址列信息达到2/3以上的匹配，同时，对 TCP 数据帧的地址列信息的识别达到 100%。因此，该方法能够有效地实现对数据帧地址列信息的识别，这可以作为将数据帧分离为点对点数据的依据。从而使协议识别获得更精确的效果，并进一步减少冗余特征。

## 参 考 文 献

[1]　官建文. 中国移动互联网发展报告[M]. 社会科学文献出版社，2012

Guan Jian-wen. Report on the development situation of China Mobile Internet[M]. Social Sciences Acadmeic Press, 2012

(下转第 211 页)

[5] 王薇. 分组密码 CLEFIA 与基于四圈 AES 的消息认证码的安全性分析[D]. 济南:山东大学,2009
Wang Wei. Cryptanalysis of Block Cipher CLEFIA and MACs based on four rounds AES[D]. Jinan: Shandong University, 2009

[6] Wang Wei, Wang Xiao-yun. Improved impossible differential Cryptanalysis of CLEFIA [EB/OL]. (2008-05-05). http://eprint.iacr.org/2007/466

[7] Tsunoo Y, Tsujihara E, Shigeri M, et al. Impossible differential Cryptanalysis of CLEFIA [C]// Proc. of FSE'08. Atlanta, USA:[s.n.],2008:398-411

[8] Zhang Wen-ying, Han Jing. Impossible differential analysis of reduced round CLEFIA[C]// Proc. of Inscrypt'08. Beijing, China,2008:181-191

[9] Tang X, Sun B, Li R, et al. Impossible differential cryptanalysis of 13-round CLEFIA-128[J]. Journal of Systems and Software, 2011,84(7):1191-1196

[10] Mala H, Dakhilalian M, Shakiba M. Impossible differential attacks on 13-round CLEFIA-128[J]. Journal of Computer Science and Technology,2011,26(4):744-750

[11] Wu Wen-ling, Zhang Lei, Zhang Wen-tao. Improved impossible differential Cryptanalysis of reduced-round Camellia[C]// Proc. of SAC'08. [S.l.]:ACM Press,2008:442-456

[12] 郑秀林,连至助,鲁艳蓉,等. CLEFIA-128 算法的不可能差分密码分析[J]. 计算机工程,2012,38(3):141-144
Zheng Xiu-lin, Lian Zhi-zhu, Lu Yan-rong, et al. Impossible differential Cryptanalysis of CLEFIA-128 algorithm[J]. Computer Engineering,2012,38(3):141-144

(上接第 187 页)

[2] 朱树永. 协议识别技术研究[D]. 长沙:国防科技大学,2008
Zhu Shu-yong. The study on protocol identification technology [D]. Changsha: National University of Defense Technology, 2008

[3] IANA[OL]. http://www.iana.org/assignments/port-umbers

[4] Liu R T, Huang N F, Chen C H, et al. A fast string-matching algorithm for network processor-based intrusion detection system [J]. ACM Transactions on Embedded Computing Systems, 2004,3(3):614-633

[5] IANA. Internet Assigned Numbers Authority [OL]. http://www.iana.org/assignments/port-numbers

[6] Kim M S, Won Y J, Hong J W K. Application-level traffic monitoring and an analysis on IP networks[J]. ETRI Journal,2005, 27(1):22-42

[7] Chen C C, Wang S D. An efficient multicharacter transition string-matching engine based on the Aho-Corasick Algorithm [J]. ACM Transactions on Architecture and Code Optimization, 2013,10(4):1-22

[8] 刘佳雄. 基于 DPI 和 DFI 技术的对等流量识别系统的设计[D]. 秦皇岛:燕山大学,2010
Liu Jia-xiong. The design for a real-time P2P traffic detection system based on DPI and DFI[D]. Qinhuangdao: Yanshan University,2010

[9] Sen S, Spatscheck O, Wang Dong-mei. Accurate, scalable in network identification of P2P traffic using application signatures [C]// Proc of the 13th International World Wide Web Conference. 2004:512-521

[10] Schiller A C, Binkley J, Harley D. Botnets: the killer Web app [M]. St Louis Mo Syngress Publishing,2006

[11] Wang Y, et al. A semantics aware approach to automated reverse engineering unknown protocols[C]// 20th IEEE International Conference on Network Protocols(ICNP 2012). Austin, TX, USA: IEEE,2012:1-10

[12] Wang Y, Zhang N, Wu Y, et al. Protocol Specification Inference Based on Keywords Identification[M]// Advanced Data Mining and Applications. Springer Berlin Heidelberg,2013:443-454

[13] Kang H J, Kim M S, Hong J W K. A method on multimedia service traffic monitoring and analysis [C]// Proc. of International Workshop on Distributed System, Operations and Management. 2003:93-105

[14] Van Der M J, Caceres R, Chu Y, et al. Mmdump: A tool for monitoring Internet multimedia traffic[J]. ACM SIGCOMM Computer Communication Review,2000,30(5):48-59

[15] 李雄伟,王希武,王盼卿. 基于模式串匹配的 Ethernet 协议识别算法研究[J]. 计算机工程与应用,2007,43(29):143-145
Li Xiong-wei, Wang Xi-wu, Wang Pan-qing. Ethernet protocol identification algorithm based on pattern matching[J]. Computer Engineering and Applications,2007,43(29):143-145

[16] 何畏,汪荣贵,查全民. 一种新的快速移动单模式匹配算法[J]. 合肥工业大学学报(自然科学版),2010,33(5):665-669
He Wei, Wang Rong-gui, Zha Quan-min. A novel fast moving algorithm for single pattern matching[J]. Journal of Hefei University of Technology(Natural Science),2010,33(5):665-669

[17] 朱姣姣,叶猛. 多模式匹配及其改进算法在协议识别中的应用[J]. 电视技术,2012,36(7):60-63
Zhu Jiao-jiao, Ye Meng. Multi-pattern Matching and Application of Improved Algorithm to Protocol Identification [J]. Video Engineering,2012,36(7):60-63

[18] 张之远,叶文晨,陈云寰. 基于多模式匹配的状态检测技术[J]. 电子测量技术,2010,33(11):98-101
Zhang Zhi-yuan, Ye Wen-chen, Chen Yun-huan. Technology of stateful inspection based on the multi-pattern matching [J]. Electronic Measurement Technology,2010,33(11):98-101

[19] 王勇,吴艳梅,李芬,等. 面向比特流数据的未知协议关联分析与识别[J]. 计算机应用研究,2015,32(1):243-248
Wang Yong, Wu Yan-mei, Li Fen, et al. Protocol identification association analysis in mobile network environment[J]. Application Research of Computers,2015,32(1):243-248

[20] 琚玉建,谢绍斌,张薇. 网络协议帧切分优化过程研究与仿真[J]. 计算机仿真,2015,32(1):318-321
Ju Yu-jian, Xie Shao-bin, Zhang Wei. Research and Simulation of Optimization Process for Network Protocol Frame Segmentation [J]. Computer Simulation,2015,32(1):318-321