面向链路比特流的未知帧关联分析

薛开平*[①] 柳 彬[①] 王劲松[②] 李 威[①] 薛颖杰[①]
[①](中国科学技术大学信息科学技术学院 合肥 230027)
[②](西南电子电信技术研究所 成都 610041)

摘 要：在电子对抗中，截获到对方的通信比特流序列之后，当链路协议类型未知时，现有的协议解析工具往往无法分析比特流所承载的有用信息。为了获取比特流承载信息，首先需要切分比特流得到链路帧。该文根据链路帧结构的一般规律，提出一种基于数据挖掘的比特流切分算法。通过频繁序列统计、关联规则分析以及关联规则整合，识别出比特流中标识帧起始的多重关联规则序列。测试结果表明，该算法能够从未知比特流中提取有效的切分标识，正确实现比特流切分。与同类基于数据挖掘的比特流分析方法相比，该算法复杂度低，输出结果唯一且可信度高。

关键词：链路比特流；未知帧；频繁统计；关联分析；切分
中图分类号：TP393 **文献标识码**：A **文章编号**：1009-5896(2017)02-0374-07
DOI: 10.11999/JEIT160289

Data Link Bit Stream Oriented Association Analysis on Unknown Frame

XUE Kaiping[①] LIU Bin[①] WANG Jinsong[②] LI Wei[①] XUE Yingjie[①]
[①](*School of Information Science and Technology*, *University of Science and Technology of China*, *Hefei* 230027, *China*)
[②](*Southwest Electronics and Telecommunication Technology Research Institute*, *Chengdu* 610041, *China*)

Abstract: In the electronic countermeasure, the opponent's bit stream can be captured. However, without any knowledge about the type of data link protocol, the existing protocol analyzing tools can not analyze the useful information from the bit stream. To further get the carried information, the bit stream should be segmented to frames firstly. According to the general rules of frame structure, a bit stream segmentation algorithm is proposed based on data mining, in which, the multi-association rule indicating the beginning of frames can be identified by using frequent sequence statistics, association analysis and association rules integration. The test results show that, this algorithm can extract the valid segmentation flag from unknown bit stream and segment the bit stream correctly. Compared to the similar data mining based bit stream analyzing algorithms, this algorithm can be more efficient and produce a unique result which is of high reliability.

Key words: Data link bit stream; Unknown frame; Frequent statistics; Association analysis; Segmentation

1 引言

目前的协议解析工具，如 Wireshark, Tcpdump，能有效完成面向已知协议的解析工作。但在日益激烈的电子对抗中，侦听者所截获的通信比特流往往协议类型完全未知，此时面向已知协议的解析工具将不起作用。为了分析截获的比特流承载信息，关键的一步首先是对其进行切分而获得链路帧，在此基础上才能进行后续解析工作。面向链路比特流的未知帧关联分析旨在寻找一种方法，从完全未知的链路比特流中寻找能指示帧起始的序列标识，基于此对比特流进行切分获得链路帧，使后续对未知帧的解析成为可能。

由于缺乏关于链路协议的先验知识，导致面向链路比特流的未知帧识别和解析困难，目前国内外相关研究较少。文献[1]基于隐式马尔科夫模型，结合报文长度和到达时间间隔两种行为模式对应用协议进行分类。文献[2]提出一种基于多模式匹配的网络视频流分类算法。文献[3]基于载荷部分字节的编码，实现对混合流量按应用类型进行有效分类。然而文献[1–3]都只适用于对网络层以上的协议进行识别分类。文献[4–7]着眼于安全协议。其中，文献[4]提出了一种深度包检测和流检测相结合的针对加密流量的应用识别算法。文献[5]基于序列模式挖掘提

收稿日期：2016-03-28；改回日期：2016-07-25；网络出版：2016-10-09
*通信作者：薛开平 kpxue@ustc.edu.cn
基金项目：国家自然科学基金(61379129)，中国科学院青年创新促进会人才基金(2016394)
Foundation Items: The National Natural Science Foundation of China (61379129), Youth Innovation Promotion Association CAS (2016394)

取协议关键词序列，并结合密文特征解析安全协议格式。文献[6]提出了一种基于主体行为特征的安全协议会话识别方法。文献[7]利用随机性检测识别比特流的加密部分和未加密部分，但该方法分类粒度较粗，无法按帧切分比特流。此外，文献[8]提出了一种基于地址信息将未知帧分离为点对点数据的方法，但前提是已将比特流正确切分成帧。文献[9]借鉴数据挖掘中频繁集和关联规则的概念[10]，但结果中存在冗余序列干扰。文献[11]可以识别标志帧起始的特征序列，但其输出中同样存在冗余序列，且在缺乏先验知识时无法验证输出是否正确。

本文基于链路帧结构的一般规律，通过频繁序列统计和关联分析获取二重关联规则集合，并利用关联规则有向图进行整合，获得多重关联规则序列，基于此进行比特流有效切分。本文的创新点包括：(1)利用有向图整合二重关联规则，得到唯一的多重关联规则作为输出结果，有效消除了输出结果的冗余；(2)提供了一种判断输出结果合理性的方法，为用户调整门限参数提供参考，确保最终结果正确可信。本文接下来的结构安排如下：第 2 节介绍未知帧关联分析的基本原理；第 3 节给出算法的具体流程；第 4 节给出算法的实际数据测试结果；最后对本文进行了总结。

2　未知帧关联分析原理

2.1 链路帧结构特征

链路帧一般由帧头、数据段、校验和帧尾 4 部分组成，如图 1 所示。某些特定类型的链路帧可能省略其中某些部分，例如 WiFi 的控制帧不含数据段。

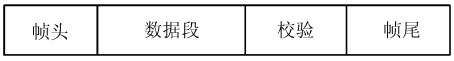

图 1 链路帧一般格式

帧格式中帧头包含帧同步序列，以及地址域、控制域等常用域。同步序列是一个特殊的模式序列，用于使接收方能从接收的比特流中区分帧的起始与终止，一种链路协议的同步序列在通信过程中始终不变。地址域、控制域等常用域的长度以及在特定帧的帧头中的位置通常是固定的，因此这些常用域之间的相对位置固定，即存在固定位置差。与同步序列类似，它们也能起到辅助界定帧起止的作用。这种帧头结构的组成规律可以概括如图 2 所示，其中固定域是帧头中内容和位置均固定的比特字段，可变域的内容在通信过程中往往会发生改变。这种

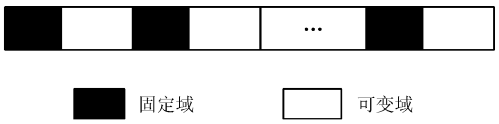

图 2 链路帧头结构

固定域和可变域间隔出现的帧头结构在各种链路帧中普遍存在，是从比特流中识别链路帧头结构的基础，寻找到帧头结构后就可以基于此将比特流切分成链路帧。

2.2 相关定义和概念

为描述链路帧头部存在的固定序列及其位置关系，首先对比特流做一些定义。

定义 1　若序列 X 包含 m 个比特位，则称 X 的长度为 m。

容易证明在长度为 n 的随机比特流 S 中，长度为 m 的序列平均出现次数为 $(n-m+1)/2^m$ [9]。

定义2　在长度为 n 的比特流 S 中，长度为 m 的序列 X 出现次数介于区间 $(\mathrm{Th}_{\mathrm{low}}\times(n-m+1)/2^m,\ \mathrm{Th}_{\mathrm{up}}\times(n-m+1)/2^m)$ 时，称 X 为频繁序列。其中 $(\mathrm{Th}_{\mathrm{low}},\mathrm{Th}_{\mathrm{up}})$ 为设定的上下限比例参数，$\mathrm{Th}_{\mathrm{low}}\in(0,1)$，$\mathrm{Th}_{\mathrm{low}}<\mathrm{Th}_{\mathrm{up}}$。

频繁序列用于描述帧头部的同步序列和常用域。在缺乏关于比特流的先验知识时，以平均出现次数 $(n-m+1)/2^m$ 为基准，其中 m 为待统计的序列长度，取值可变，理论上认为出现次数大于该值的序列是“频繁”的，但为了不过早排除帧头部的有关序列，实际操作时一般取 $\mathrm{Th}_{\mathrm{low}}<1$。设置上限参数是为了限制参与后续分析的频繁序列数量，有利于提高算法效率。

定义 3　设 X，Y 为频繁序列，$\mathrm{pos}(X)$，$\mathrm{pos}(Y)$ 为 X，Y 在比特流中的位置，$\mathrm{supp}(X)$，$\mathrm{supp}(Y)$，$\mathrm{supp}(\mathrm{pos}(X)-\mathrm{pos}(Y)=C)$ 分别表示比特流中序列 X，Y 以及 X 与 Y 间隔 C 比特出现的次数，则关联规则 $X\rightarrow Y$ 的置信度为

$$\mathrm{Conf}(X\rightarrow Y)=\frac{\mathrm{supp}\big(\mathrm{pos}(X)-\mathrm{pos}(Y)=C\big)}{\min\big(\mathrm{supp}(X),\mathrm{supp}(Y)\big)} \quad (1)$$

定义 4　设 T 为关联规则置信度门限，若 X 与 Y 间隔 C 出现的置信度满足

$$\mathrm{Conf}(X\rightarrow Y)\geq T \quad (2)$$

则称 X 与 Y 之间存在位置差为 C 的关联规则，X 称为关联规则先导，Y 称为关联规则后继。关联规则用于描述帧头部固定序列间位置差固定的关系。

3　未知帧切分算法

本文所提出的未知帧切分算法包括 4 个步骤：(1)设定频繁度门限区间，从比特流中筛选出满足频

繁度要求的序列，然后两两考察频繁序列，寻找序列间的关联规则；(2)为充分考虑关联规则间的联系，将得到的所有关联规则按一定方式组织成有向图；(3)为有向图的序列节点分配坐标，整合参与形成有向图的二重关联规则，获得多重关联规则；(4)根据多重关联规则在比特流中出现的频率判断其合理性，根据判断结果调整频繁度门限区间，确保获得的结果正确可信。

3.1 频繁统计与关联分析

频繁统计通过对比特流中的序列计数，筛选出满足频繁度区间要求的序列参与后续关联分析[12]。传统单模式匹配算法[13,14]一次只能寻找一个特定模式序列，由于待统计的序列长度不一，每种长度又对应多个序列，利用多模式匹配算法[15,16]能够只扫描一次源数据找到多个模式，更适合比特流中的序列统计。结合比特流特征，将多模式匹配 AC 算法[17]稍作改进，可有效面向比特流进行频繁序列统计，扫描一次源数据即可得所有长度比特序列的计数[9]，再从中选出满足门限要求的序列即可。

在链路帧承载的上层数据未知时，可假定帧头之外的帧数据段比特流是随机的。通常在链路帧中数据段远长于帧头，数据段大量的随机比特会干扰帧头固定序列的寻找。通过频繁统计得到的比特序列多数来自数据段，不能作为帧分界标识。根据对帧结构特征的分析可知，帧头固定序列间存在位置差固定的关联规则。由于数据段比特流具有随机性，来自数据段的序列成为频繁序列存在偶然性，且它们在比特流中出现的位置具有随机性，这种“伪频繁序列”之间存在关联规则的概率很低。因此可以认为，比特流中序列间位置差固定的关联规则一般来自帧头。

验证序列之间的关联规则需要首先计算序列位置差，对于任意两个频繁序列 X 和 Y，相对位置差为 C 有 $X \to Y$ 和 $Y \to X$ 两种可能的顺序，必须对两种情况都进行验证。验证时取 X 和 Y 中出现次数较少者作为基准，假定为 X，对每一个 X，计算与其距离最近且在其前面出现的 Y 的位置差 $\mathrm{pos}(Y)-\mathrm{pos}(X)$，并确定是否有常数 C 满足关联规则置信度要求；再计算与其距离最近且出现在其后面的 Y 的位置差 $\mathrm{pos}(X)-\mathrm{pos}(Y)$，并确定是否有常数 C 满足关联规则置信度要求[9]。验证流程如图 3 所示。

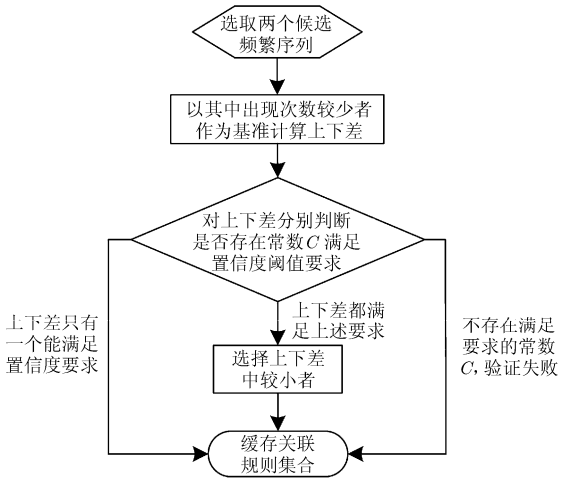

图 3 关联规则验证流程

通过关联规则验证所得的是由若干关联规则组成的集合。在缺乏先验知识时，用户对集合中的关联规则依然无从挑选。为解决此问题，需要对集合中的关联规则进一步处理以获得唯一可信的输出。

3.2 关联规则有向图的生成

实验中发现，通过 3.1 节获得的集合中的关联规则之间会存在 3 种类型的联系：

(1)一条关联规则的后继是另一条关联规则的先导；

(2)两条关联规则存在公共的先导，此时两条关联规则的后继可能存在公共子序列；

(3)两条关联规则存在公共的后继，此时两条关联规则的先导可能存在公共子序列。

图 4 所示是关联规则之间 3 种联系的具体示例：图 4(a)中序列 V_1 既是关联规则 $V_0 \to V_1$ 的后继，又是关联规则 $V_1 \to V_2$ 的先导；图 4(b)中关联规则 $V_1 \to V_2$ 与 $V_1 \to V_3$ 存在公共的先导 V_1，同时两条关联规则的后继 V_2 和 V_3 与先导 V_1 间隔距离相差 4 bit，V_2 的末尾与 V_3 的开头存在长度为 4 bit 公共子序列 0111；图 4(c)中关联规则 $V_0 \to V_1$ 与 $V_4 \to V_1$ 存在公共的后继序列 V_1，同时两条关联规则的先导 V_0 和 V_4 与后继 V_1 间隔距离相差 3 bit，V_0 的开头与 V_4 的末尾存在长度为 5 bit 公共子序列 00001。

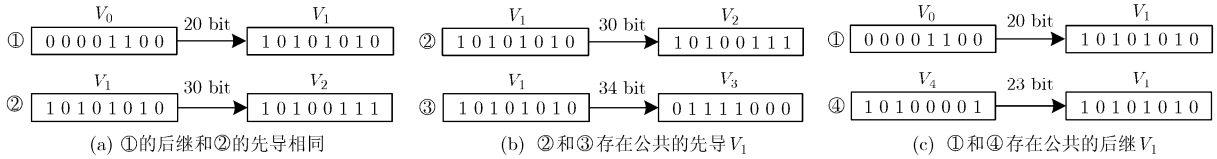

图 4 关联规则之间的 3 种联系

关联规则之间的这些联系提供了通过生成有向图对关联规则进行整合的基础。

为充分发现集合中关联规则之间的联系，可将关联规则按如下方式组织成带权重有向图：(1)两个序列充当节点；(2)序列的前后顺序决定有向图边的方向；(3)两个序列位置的间隔比特数充当有向边的权重。按照此方式，作为示例，将图 4 中 4 条关联规则组织成如图 5 所示的有向图。

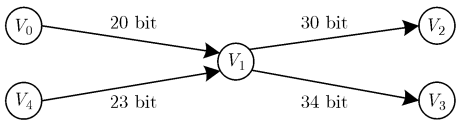

图 5 关联规则有向图

3.3 由有向图生成多重关联规则

有向图生成后，为整合各关联规则，获得能作为比特流切分标识的多重关联规则，需要首先确定有向图各节点序列在多重关联规则中的相对位置。为了确定各节点序列的相对位置，可以将节点序列的第一个比特在多重关联规则中出现的位置作为节点坐标，通过节点坐标间的差值表示节点的相对位置。为了计算节点序列坐标，首先将有向图用邻接矩阵形式表示。设 $G=(V,E)$ 是有 n 个顶点的有向图，其中 V 表示顶点集合，E 表示边的集合，则 G 的邻接矩阵可定义为有如下性质的 n 阶方阵：

$$\boldsymbol{A}[i,j]=\begin{cases}W_{ij}, & (V_i,V_j)\in E\\ -1, & (V_i,V_j)\notin E \text{ 或 } i=j\end{cases} \tag{3}$$

其中 W_{ij} 为有向边 (V_i,V_j) 的权重。设图 5 有向图的邻接矩阵为 $\boldsymbol{A}_1$，按定义有

$$\boldsymbol{A}_1=\begin{bmatrix}-1 & 20 & -1 & -1 & -1\\ -1 & -1 & 30 & 34 & -1\\ -1 & -1 & -1 & -1 & -1\\ -1 & -1 & -1 & -1 & -1\\ -1 & 23 & -1 & -1 & -1\end{bmatrix} \tag{4}$$

节点坐标只有相对意义，在计算坐标前，可任选一节点，为其坐标赋值 0，再结合邻接矩阵即可计算出其余节点序列坐标。节点序列坐标计算过程如表 1 所示。

初始化图 5 中节点 V_1 的坐标为 0，根据表 1 计算图 5 各节点坐标如表 2 所示。

根据节点序列坐标，可确定各节点序列在未知帧头部的顺序。实验中发现，计算坐标时会出现两序列坐标之差小于前一序列长度的情况。这是由于两序列存在公共子序列，此时需将存在公共子序列的两序列合并。如表 2 中 V_4 =10100001，V_0 =00001100，坐标

表 1 有向图各节点坐标的计算

输入：关联规则带权有向图的邻接矩阵。

输出：有向图所有节点的坐标。

(1)for(有向图的每一个节点 V_i)

{将 V_i.flag 置 0; //V_i.flag = 0 表示节点 V_i 的坐标尚未确定；}

(2) V_0.coor $\leftarrow 0$, V_i.flag $\leftarrow 1$, sum $\leftarrow 1$, $i\leftarrow 1$; //初始化 V_0 坐标为 0, sum 记录当前已确定坐标的节点数目，i 为当前待计算坐标的节点的序号；

(3) While(sum <有向图节点数)

{if(节点 V_i 的某一相邻节点 V_j 的坐标已确定)

{根据 V_i.coor=$\begin{cases}V_j.\text{coor}+\text{length}(V_j)+W_{ij}, & W_{ij}\neq -1\\ V_j.\text{coor}-\text{length}(V_j)-W_{ji}, & W_{ji}\neq -1\end{cases}$

计算 V_i 坐标，其中 length(V_j) 表示节点序列 V_j 长度；

V_i.flag $\leftarrow 1$; sum + + ;}

i + + ;

if(i =有向图节点数) $i\leftarrow 1$;

}

(4) 输出所有节点的坐标。

表 2 节点坐标计算结果

节点序列	V_4	V_0	V_1	V_2	V_3
坐标	−31	−28	0	38	42

分别为−31, −28，两者坐标之差为 3，小于 V_4 的长度，V_4 末尾与 V_0 开头存在公共子序列 00001，这时将二者合并成序列 10100001100，并将合并所得序列坐标设为 V_4 (两者中坐标较小者)坐标即可。

合并完成后，根据合并后序列的长度和序列坐标计算相邻序列的位置间隔，即可得多重关联规则。由表 2 中的节点坐标计算结果，进行序列合并后所得多重关联规则如图 6 所示。

3.4 多种关联规则的合理性判断

获得多重关联规则后，可判断其合理性：若未知比特流中包含 k 条链路帧，则帧头在比特流中也会出现 k 次。通过 3.1 节的关联分析所得集合中的关联规则来自帧头，因此它们出现的次数一般会与

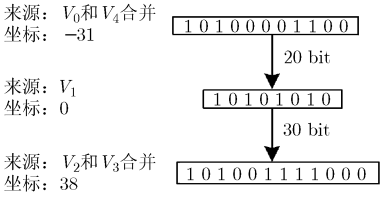

图 6 多重关联规则

k 相等，受帧数据段偶然出现的关联规则干扰，部分关联规则出现次数可能略大于 k 。通过对 3.1 节所得集合中各关联规则出现次数，即式(1)中的 $\mathrm{supp}(\mathrm{pos}(X)-\mathrm{pos}(Y)=C)$ 取平均值来估测 k 值，设所得平均值为 k_0 。再统计所得多重关联规则在比特流中出现的次数，设为 k_1 。若 $k_1 \approx k_0$，则认为结果合理；若 k_1 与 k_0 相差过大($k_1 \ll k_0$)，则结果不可信，需调整 $(\mathrm{Th}_{\mathrm{low}}, \mathrm{Th}_{\mathrm{up}})$ 重新分析。

经过判断确定多重关联规则合理后，即可将其作为界定帧起始的标识切分比特流。

4 实验

本文采用实际数据对算法进行测试，使用 Qt Creator 3.1.0 平台实现所提出的算法，硬件配置为：Intel i5-2450M, 2.5 GHz，双核 CPU, RAM 4 GB。

4.1 算法有效性验证

为验证算法的有效性，实验采用两组 Wireshark 实际抓获的数据集：Ethernet 链路层数据，共 1000 个 Ethernet 帧，大小为 1075 kb; WiFi 链路层数据，共 500 个 WiFi 帧(382 个数据帧，118 个管理帧)，大小为 458 kb。为方便测试，测试中只统计长为 8 bit 的频繁序列，更长序列可以在此基础上通过序列合并获得。在 1075 kb 的 Ethernet 数据和 458 kb 的 WiFi 数据中，长 8 bit 的序列出现的平均次数分别为 4300 和 1830。$(\mathrm{Th}_{\mathrm{low}}, \mathrm{Th}_{\mathrm{up}})$ 分别设定为(0.50,0.75)和(0.90,1.20)，此时频繁度门限分别为 2150–3225 和 1647–2196，置信度门限设为 0.3。

实验结果如表 3 所示。可以看出，在数据量较大时，算法能够从中寻找到正确的比特流切分标识。

表 3 Ethernet 和 WiFi 数据测试结果

协议类型	Ethernet		WiFi	
是否找到帧头结构	是		是	
数据集大小	1075 kb		458 kb	
算法输出	序列	坐标	序列	坐标
	“10101011”	−118	“0101110011”	−106
	“10110100”	−81		
	“11110000”	−66	“11010100”	−14
	“1010011110111110100”	−44		
	“01001001”	−23	0x2E444FDF	0
	“00010001010”	0		

在原数据中进一步分析输出序列，以 Ethernet 数据集输出结果为例，确定输出序列在原数据帧中的相对位置及承载的含义，结果如表 4 所示：Ethernet 数据集输出结果为 6 个长度不等的序列之间间隔若干比特的多重关联规则。最前面的序列“10101011”是 Ethernet 帧的特征序列，用于标识一个 Ethernet 帧的开始，后面的 4 个序列属于 Ethernet 帧头部源和目的 MAC 地址字段，最后一个序列属于 Ethernet 的 IP 负载首部 IP 版本号和长度字段。中间间隔比特位属于帧头部 MAC 地址字段，由于频繁度门限设置的原因被排除在频繁序列之外。基于这种来自 Ethernet 帧头的多重关联规则，方案可正确切分 Ethernet 比特流。

4.2 上下限比例参数对算法的影响

在算法设计中，频繁统计涉及上下限比例参数 $(\mathrm{Th}_{\mathrm{low}},\ \mathrm{Th}_{\mathrm{up}})$ 的设置，为考察 $(\mathrm{Th}_{\mathrm{low}},\ \mathrm{Th}_{\mathrm{up}})$ 设置对算法的影响，实验采用 1075 kb 的 Ethernet 数据集，调整频繁度门限观察实验结果。实验结果如表 5 所示，为简化表格，未列出输出结果的具体内容，而是用 (n,m) 的形式表示输出结果的长度信息，其中 m 表示关联规则序列总长度，n 表示关联规则中已明确挖掘出的比特序列长度。

表 4 算法输出结果承载含义

比特位	值	承载内容
1~8	10101011	前导码
9~37	未挖掘出的序列	MAC 地址
38~45	10110100	MAC 地址
46~52	未挖掘出的序列	MAC 地址
53~60	11110000	MAC 地址
61~74	未挖掘出的序列	MAC 地址
75~92	1010011110111110100	MAC 地址
93~95	未挖掘出的序列	MAC 地址
96~103	01001001	MAC 地址
104~118	未挖掘出的序列	MAC 地址
119~129	00010001010	IP 头部字段

表 5 频繁度门限区间对算法的影响

上下限比例参数	输出结果长度	是否找到帧头结构	算法耗时(s)
(0.50, 0.60)	(16, 47)	是	44.5
(0.50, 0.65)	(16, 47)	是	58.9
(0.50, 0.70)	(48, 129)	是	130.3
(0.50, 0.75)	(61, 129)	是	220.8
(0.50, 0.80)	(86, 101)	否	530.3

从表 5 可以看出，随着上下限比例参数差值增加，算法输出的多重关联规则序列长度增加，方案所耗时间增长。这是因为增大区间长度后，筛选得到的频繁序列更多，而方案所耗时间主要来自两两考察序列之间的关联规则，这部分的时间复杂度为 $O(n^2)$ (n 为频繁序列数目)。上下限比例参数差值增大后，结果可能包含错误信息。因为在上下限比例参数差值增大后，筛选得到的频繁序列更多，其中某些频繁序列可能在帧头不同位置出现多次，为不同位置的同一序列分配一个坐标导致结果出错。根据 3.4 节的合理性判断可发现这种错误，发现后适当缩小上下限差值即可。

为节省计算时间，在保证结果正确性的前提下，应选取尽可能短的频繁度门限区间。最短门限区间的具体取值(起点、终点)会受待分析的链路比特流数据的影响。在实际应用时，可结合 3.4 节的合理性判断，经过多次测试确定最短门限区间取值。根据实验经验，初次设置可取 0.5 作为门限区间中点，再经测试调整确定区间的具体取值。

4.3 与其他算法比较

文献[11]提出的比特流分割算法，输出结果是按出现次数降序排列的 N 个关联规则。采用 4.1 节数据集，在设置关联规则置信度为 0.9 时，表 6 列出了算法输出中排名前 5 的关联规则序列(表中?表示间隔比特)。通过对比原数据分析，在 Ethernet 对应的输出结果中，第 1 位即为正确切分标志；WiFi 数据对应输出的前 5 位中不含正确切分标志，正确标志最早出现在输出结果的第 17 位。在 N 较大时，文献[11]能保证输出结果中包含正确比特流切分标志，但正确标志在输出中位置不定，缺乏先验知识时，用户对输出结果无从选择；在 N 较小时，算法输出的 N 个关联规则中可能不包含正确的切分标识，结果不可信。本文算法能提供唯一输出结果，有效消除输出中的冗余；同时根据 3.4 节的合理性判断与参数调整，可确保最终结果正确可信。与文献[11]相比，本文算法输出结果唯一且可信度更高。

表 6 文献[11]算法输出结果

协议类型	Ethernet	WiFi
输出前 5 位	1 : 10111101??0110100100110000	1 : 0001100001000110000011101
	2 : 11011110???0110100100110000	2 : 0011000010001100000111010
	3 : 10101100?????????1000001010110111	3 : 0000110000100011?00011101
	4 : 0101100000000001010000010	4 : 0000001100001000110000011
	5 : 00101001?????????????????????????0110000100000000	5 : 0010001100001110100011101
正确标志最小排序	1	17

在与 4.1 节相同的数据集上，本文算法和文献[11]算法进行了运算时间的比较测试，实验结果如表 7 所示。由于算法的复杂度主要来自两两考察频繁序列之间的关联规则，文献[11]认为超过平均次数 $(n-m+1)/2^m$ 的序列均为频繁序列，未设置频繁度上限，需要考察的频繁序列数量超过本文算法，对相同的链路比特流数据，本文算法的处理效率高于文献[11]算法。

以上实验结果表明，对帧头部存在位置差固定的关联规则的链路比特流，在设定合理门限后，算法能在相对较短的时间内获得有效的比特流切分标识，实现比特流切分。

表 7 不同算法运算时间比较(s)

数据(大小)	文献[11]算法	本文算法
Ethernet(1075 kb)	2339	220
WiFi(458 kb)	236	28

5 结束语

对链路层截获的比特流数据，如何在缺乏先验知识的情况下获取其中承载的有用信息是一项重要的研究课题，分析有用信息的第 1 步要求正确切分比特流提取链路帧。本文分析了一般链路帧的帧格式特征，结合数据挖掘的关联分析方法，从未知比特流中寻找能作为未知帧切分标识的多重关联规则，基于此对未知比特流进行有效切分。最后结合实际数据，通过实验测试了方案的有效性，实验结果证明方案能有效寻找到比特流中标识帧起始的定界序列，为后续链路层数据的解析工作提供了支撑。当然方案还有改进的地方，例如频繁门限区间目前还需要多次测试后才能确定，如何自适应确定最佳

频繁门限区间需要进一步研究；在切分比特流之后，如何在缺乏先验知识的情况下解析链路帧承载的信息也是下一步研究的重点。

参 考 文 献

[1] WRIGHT C, MONROSE F, and MASSON G M. HMM profiles for network traffic classification[C]. Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security. ACM, Washington, D.C., USA, 2004: 9–15. doi: 10.1145/1029208.1029211.

[2] 孙钦东, 郭晓军, 黄新波. 基于多模式匹配的网络视频流识别与分类算法[J]. 电子与信息学报, 2009, 31(3): 759–762. doi: 10.3724/SP.J.1146.2008.00301.
SUN Q, GUO X, and HUANG X. Algorithm of network video stream recognition and classification based on multi-pattern matching[J]. *Journal of Electronics & Information Technology*, 2009, 31(3): 759–762. doi: 10.3724/SP.J.1146.2008.00301.

[3] 王变琴, 余顺争. 未知网络应用流量的自动提取方法[J]. 通信学报, 2014, 35(7): 164–171. doi: 10.3969/j.issn.1000-436x.2014.07.020.
WANG B and YU S. Automatic extraction for the traffic of unknown network applications[J]. *Journal on Communications*, 2014, 35(7): 164–171. doi: 10.3969/j.issn.1000-436x.2014.07.020.

[4] 高长喜, 吴亚飚, 王枞. 基于抽样分组长度分布的加密流量应用识别[J]. 通信学报, 2015, 36(9): 65–75. doi: 10.11959/j.issn.1000-436x.2015171.
GAO C, WU Y, and WANG C. Encrypted traffic classification based on packet length distribution of sampling sequence[J]. *Journal on Communications*, 2015, 36(9): 65–75. doi: 10.11959/j.issn.1000-436x.2015171.

[5] 朱玉娜, 韩继红, 袁霖, 等. SPFPA: 一种面向未知安全协议的格式解析方法[J]. 计算机研究与发展, 2015, 52(10): 2200–2211. doi: 10.7544/issn1000-1239.2015.20150568.
ZHU Y, HAN J, YUAN L, *et al*. SPFPA: A format parsing approach for unknown security protocols[J]. *Journal of Computer Research and Development*, 2015, 52(10): 2200–2211. doi: 10.7544/issn1000-1239.2015.20150568.

[6] 朱玉娜, 韩继红, 袁霖, 等. 基于主体行为的多方安全协议会话识别方法[J]. 通信学报, 2015, 36(11): 190–200. doi: 10.11959/j.issn.1000-436x.2015273.
ZHU Y, HAN J, YUAN L, *et al*. Towards session identification using principal behavior for multi-party secure protocol[J]. *Journal on Communications*, 2015, 36(11): 190–200. doi: 10.11959/j.issn.1000-436x.2015273.

[7] 邢萌, 王韬, 吴杨, 等. 一种提高链路层加密比特流识别率的新方法[J]. 计算机应用研究, 2015, 32(11): 3443–3447. doi: 10.3969/j.issn.1001-3695.2015.11.057.
XING M, WANG T, WU Y, *et al*. New method to improve identification rate of encrypted bit stream in data link layer[J]. *Application Research of Computers*, 2015, 32(11): 3443–3447. doi: 10.3969/j.issn.1001-3695.2015.11.057.

[8] 郑杰, 朱强. 未知单协议数据帧的地址分析与研究[J]. 计算机科学, 2015, 42(11): 184–187. doi: 10.11896/j.issn.1002-137X.2015.11.038.
ZHENG J and ZHU Q. Analysis and research on address message of unknown single protocol data frame[J]. *Computer Science*, 2015, 42(11): 184–187. doi: 10.11896/j.issn.1002-137X.2015.11.038.

[9] 金凌. 面向比特流的未知帧头识别技术研究[D]. [硕士论文], 上海交通大学, 2011.
JIN L. Study on bit stream oriented unknown frame head identification[D]. [Master dissertation], Shanghai Jiao Tong University, 2011.

[10] WU X, ZHU X, WU G Q, *et al*. Data mining with big data[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2014, 26(1): 97–107. doi: 10.1109/TKDE.2013.109.

[11] 王和洲, 薛开平, 洪佩琳, 等. 基于频繁统计和关联规则的未知链路协议比特流切割算法[J]. 中国科学技术大学学报, 2013, 43(7): 554–560. doi: 10.3969/j.issn.0253-2778.2013.07.006.
WANG H, XUE K, HONG P, *et al*. An unknown link protocol bit stream segmentation algorithm based on frequent statistics and association rules[J]. *Journal of University of Science and Technology of China*, 2013, 43(7): 554–560. doi: 10.3969/j.issn.0253-2778.2013.07.006.

[12] AGRAWAL R, IMIELINSKI T, and SWAMI A. Mining association rules between sets of items in large databases[C]. Proceedings of ACM SIGMOD International Conference on Management of Data. Washington, D.C, USA, 1993: 207–216. doi: 10.1145/170036.170072.

[13] KNUTH D E, MORRIS,J J H, and PRATT V R. Fast pattern matching in strings[J]. *SIAM Journal on Computing*, 1977, 6(2): 323–350. doi: 10.1137/0206024.

[14] BOYER R S and MOORE J S. A fast string searching algorithm[J]. *Communications of the ACM*, 1977, 20(10): 762–772. doi: 10.1145/359842.359859.

[15] HONG Y D, KE X, and YONG C. An improved Wu-Manber multiple patterns matching algorithm[C]. IEEE Performance, Computing and Communications Conference, Phoenix, Arizona, USA, 2006: 674–680. doi: 10.1109/.2006.1629469.

[16] FAN J J and SU K Y. An efficient algorithm for matching multiple patterns[J]. *IEEE Transactions on Knowledge and Data Engineering*, 1993, 5(2): 339–351. doi: 10.1109/69.219740.

[17] AHO A V and CORASICK M J. Efficient string matching: an aid to bibliographic search[J]. *Communications of the ACM*, 1975, 18(6): 333–340. doi: 10.1145/360825.360855.

薛开平： 男，1980年生，副教授，研究方向为下一代Internet网络、网络安全与分布式网络.

柳 彬： 男，1991年生，硕士生，研究方向为网络安全与协议分析.

王劲松： 男，1980年生，工程师，研究方向为大数据挖掘和数据可视化.

李 威： 男，1992年生，硕士生，研究方向为网络安全协议设计与分析.

薛颖杰： 女，1992年生，硕士生，研究方向为网络安全和加密技术.