TPCAD：一种文本类多协议特征自动发现方法

赵咏[1,2,3]，姚秋林[1,3]，张志斌[1,3]，郭莉[1,3]，方滨兴[1,3]

(1. 中国科学院 计算技术研究所，北京 100190；2. 中国科学院 研究生院，北京 100049；
3. 信息内容安全技术国家工程实验室，北京 100049)

摘 要：流量分类在深度包检测等网络信息安全领域具有广泛应用，而协议特征的发现是流量分类中的一个重要问题。基于文本类协议的特点提出了一种准确、高效的多协议特征自动提取方法。利用网络流量中文本内容的语义特点，将流量解析成语义单位，提出了一个在语义空间上的相似性比较方法，并据此对文本类的流量进行聚类。然后合并同一类的网络流量并提取出同类流量所共有的特征。实验表明，该方法对常见文本类协议拥有超过 95% 的准确率，并可以实现对流量的在线学习和分类。

关键词：协议特征；流量分类；语义；聚类

中图分类号：TP393 **文献标识码**：A **文章编号**：1000-436X(2009)10A-0028-08

TPCAD: a text-oriented multi-protocol inference approach

ZHAO Yong[1,2,3], YAO Qiu-lin[1,3], ZHANG Zhi-bin[1,3], GUO Li[1,3], FANG Bin-xing[1,3]

(1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China;
2. Graduate School of Chinese Academy of Sciences, Beijing 100049, China;
3. National Engineering Laboratory for Information Security Technologies, Beijing 100049, China)

Abstract: Protocol inference, which discovers protocol characteristics automatically, is an important problem in traffic classification. An accurate and efficient inference method based on semantic analysis of text traffic was proposed. According to the semantic analysis, the text content of traffic to token sequences was resolved. Then the token sequences based on a similarity comparison criterion on semantic space was clustered. At last the recognition characteristics was extracted from the token sequences that belong to the same protocol. In the experiment, the precision ratio is above 95% for common protocols and the speed can meet the needs of online training and classification.

Key words: protocol inference; traffic classification; semantic; clustering

1 引言

流量分类根据在流量中所发现的特征来区分不同的流量，被广泛用于深度包检测等网络安全领域。一般在网络安全监测系统中，首先根据协议特征对流量进行划分，然后利用深度包检测技术对流量进行深度处理，识别病毒、垃圾邮件等有害信息。最传统和直接的流量分类方法是通过流量的通信端口来推测流量所属协议及内容。基于端口的方法根据 IANA(国际互联网代理成员管理局)[1]定义的端口分配规则来区分不同的流量，但是这个方法目前面临 2 个问题：首先，大量新出现的网络应用不再遵守 IANA 的约定；其次，部分新的应用采用各种技术逃避流量分类算法。当前流行的网络应用绝

收稿日期：2009-08-10
基金项目：国家重点基础研究发展计划（“973”计划）基金资助项目（2007CB311100）
Foundation Item: The National Basic Research Program of China (973 Program) (2007CB311100)

大部分提供端口自定义功能，用户可以自己定义非标准的端口来逃避基于端口的流量分类算法。不仅如此，一些应用在防火墙禁止绝大部分通信端口时可以自动使用未被禁止的常用端口进行通信。因此，基于端口的方法会慢慢地失去作用。

为了克服基于端口方法的缺陷，许多新的方法提了出来。这些方法可以分为 2 类：一种是通过流量外部特征来进行区分的基于行为的方法；一种是通过流量内部具体内容来进行区分的基于负载的方法。

基于行为的方法采用统计学习的方法对包到达时间间隔、包大小、包数目等统计特征进行学习。这种方法假设相同协议的流量具有类似的统计特征，根据这个特征可以区分不同的协议流量。这种方法的最大好处是不需要观察包的具体内容，有助于保护隐私和提高处理速度。但是由于特征信息的稀少和不稳定，分类结果容易受到其他因素的干扰。

基于负载的方法根据包的内容特征来区分不同的流量，准确性最好。L7-filter[2]和 Wireshark[3]都是使用基于负载方法的流量分类和流量分析工具。但是这种方法的最大问题是流量特征的获取方法，目前比较常用的方法是通过手工分析获得，如 L7-filter 和 Wireshark 都是这样获得流量特征的。随着网络应用的日益增多和协议格式的复杂化，手工分析已经不能满足当前的需要，于是有人开始研究自动化的协议逆向工程方法。Cui[4]将流量分为文本和二进制 2 种类型，文本类流量是指主要由可打印字符组成的流量，其他为二进制类流量。Cui 的方法能够推测出大部分的协议格式，但是这种方法只能处理一种协议的流量，所以要求流量主要由一种协议组成，对流量的获取要求较高。Ma[5]提出可以根据服务器 IP 和端口的不同对混杂的流量进行预聚类，然后再提取每个聚类的协议特征。这样可以自动发现多协议的特征。广域网环境中存在大量使用随机端口通信的协议流量，这影响了 Ma 方法的使用效果，于是本文提出了一个适合广域网环境的文本类多协议特征自动发现方法(TPCAD，text protocol characteristic automatic discoverer)。TPCAD 算法对流量的文本内容进行处理，根据文本内容的语义关系将流量进行切割。算法比较流量中相同位置的语义块的类型和值，依据相似性的大小进行合并。最后将流量分成几个集合，每个集合的流量都属于同一个协议。对于每一个流量集合，算法以语义块为单位寻找流量的特征，归并同一位置上相似的语义块，从而抽象出该协议的负载特征。通过实验可以看出 TPCAD 方法对常见协议的识别错误率小于 3%，并且除 Gnutella 之外其他协议的召回率均达到 90%以上。在处理速度方面，骨干网半小时数据的训练时间只需 47s，分类只需 5s，完全可以适应在线分类的要求。

本文的组织如下：第 2 节介绍了本方向的相关工作，第 3 节详细阐述了 TPCAD 方法的具体过程，第 4 节通过实验测试了算法的性能，第 5 节是结束语。

2　相关工作

Claffy 在他的博士论文[6]中全面讨论了流量分类的相关问题。之后出现了使用机器学习方法对行为特征进行学习的流量分类方法，Roughan 等人[7]的工作作为其中的代表，使用最近邻（NN）和线性判别分析（LDA）方法在不同的应用程序数据和 QoS 类别之间建立映射关系。基于行为的方法需要解决的最大问题是如何寻找有效的行为特征和提高准确度。Moore[8]等人基于有监督的 Navie Bayes 算法对包括包长，包间隔以及其他 TCP 头信息在内的 248 个流特征进行训练，然后通过关联特征选择算法对特征进行筛选，最后发现仅需一小于 20 个的特征子集就可以获得比较正确的分类。Karagiannis 等人[9]将 3 个不同层次上的行为特征综合起来进行流量分类，从而达到了接近于基于负载方法的分类精度。基于负载的流量分类方法目前主要集中在对负载特征的自动发现的研究上。Weidong Cui 等人在文献[4]中设计了一个 Discoverer 系统，利用提出的聚类方法能够对属于同一协议的输入数据进行逆向工程以提取协议字段语义信息。Cui 的方法将流量区分为文本和二进制 2 种，在聚类时主要考虑二进制部分的语义信息，最后可以推测出协议绝大部分的格式信息。但事实上只需要流量的开头部分就可以实现流量分类。Haffner 等[10]发现了只需要前 64Bytes 负载就可以很好地发现特征并进行流量分类。基于 Haffner 等人的发现，Justin Ma 等人在文献[5]中，提出了 3 种协议推理模型，分别是基于统计的乘积分布模型(production distribution model)、马尔科夫过程模型(Markov process model)和通用子串图模型(common substring graph)。Justin Ma 的 3 种协议推理的优点

是无需标注数据，可对原始多协议混杂并存的网络流数据进行聚类分析并输出协议特征。在 Justin Ma 等人自己提供的训练和测试数据下，上述 3 个模型取得了很好的识别效果。

3 协议特征的自动发现

3.1 方法概述

Ma 按照服务器 IP 和服务器端口来区分不同的协议，这是一个对基于端口方法的扩展。这样的方法在局域网中可以达到较好的效果，但是仍然具有基于端口方法通病。随着随机端口技术的广泛使用，Ma 的预分类方法会逐渐失去效果。TPCAD 方法不再使用端口作为预分类的标准，它将网络中的流根据语义信息解析成 Token 序列，然后根据语义的相似性进行预分类，将不同协议的流量分到不同的集合之中。随后对单一协议的流量集合进行格式推测。TPCAD 方法只考虑文本流量的语义特征，因此目前只能发现文本流量的负载特征。图 1 为 TPCAD 方法的处理过程。

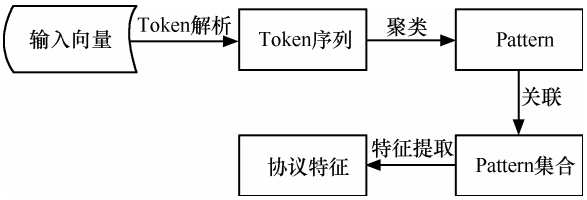

图 1 特征提取过程

3.2 预分类

预分类过程分为 Token 解析、聚类、关联 3 部分。首先流量被解析成 Token 序列并按照相似性进行聚类。然后对于每一类的 Token 序列都进行合并得到 Pattern，接着将 Pattern 按连接关系进行关联得到 Pattern 集合。

3.2.1 数据准备

互联网采用包机制进行数据传输，而本文的处理对象是流。流是指 2 台主机之间持续传输的数据。2 台主机之间的通信过程叫作一个会话，一个会话包含 2 条不同方向的流。为了重现流内容，将同一流中的数据包进行合并，然后保留流的内容。根据 Haffner 等人的发现[6]，每条流只保存前 64Bytes 负载内容。TPCAD 方法目前只处理文本类协议数据，本文目前对文本类协议数据和非文本类协议数据的分类依据是：首先统计 64Byte 的数据中可打印字符的总数以二进制块总数，如果可打印字符总数大于总字符数的 50%，而且二进制块总数小于 10，那么认为当前数据为文本类协议数据，否则为非文本类协议数据。最后将会话的四元组、流方向和负载内容保存为输入向量。

3.2.2 语义解析

由于网络流量中存在大量负载重复的输入向量，TPCAD 方法首先合并负载相同的输入向量。为了在合并的同时保留相应的四元组信息，将一个四元组列表与输入向量相关联，在今后 Token 序列、Pattern 的合并过程中四元组列表都一同进行合并操作。

随后算法将空白字符作为分隔符对输入向量进行分割，分割后的每一部分叫作一个 Token，而输入向量则变成了 Token 序列。按是否为文本可以将 Token 分为文本类和非文本类。非文本类为二进制内容，后面仅对其值进行比较不再做进一步的处理。文本类的 Token 还可以进一步划分，按照语义特征可以分为句子、版本号、URL 和普通文本 4 种类型。语义解析的任务就是识别出这几种类型的 Token。Token 不仅具有类型，还具有属性：固定和可变的。版本号和 URL 只能是可变属性的，可变属性的 Token 在 Token 合并后也可能出现，具有不同的取值。

1) URL 的解析：URL 是在网络协议中经常出现的一种格式化字符串，具有明显的语义信息和特征。图 2 是一个典型的包含 URL 的网络流。可以看出 URL 在流中具有明显的实际含义而且趋向于出现在固定的位置。不过根据 URL 的 RFC 定义[11]可知解析与定位 URL 十分复杂，因此提出了一个近似的 URL 解析方法。将流量中常见的 URL 分为 3 种形式：包含域名的 URL、包含 IP 地址的 URL 和以“/”开头的 URL。对于包含域名的 URL，根据顶级域名列表寻找域名字符串，然后在扩展至整个 URL 串。其他 2 种 URL 也采取类似的方法进行解析。

```
220 smtp1.tessera.com modusGate ESMTP Receiver Version 4.4.568.3
```

图 2 典型的文本类协议流量

2) 版本号的解析：版本号也经常出现在各种文本类的网络协议中。它是一个以“.”分隔开的一串数字，这与 IP 地址很容易混淆。因此首先判断数字串是否为 IP 地址，不是的则为版本号。

3) 普通文本和非文本的解析：根据 ASCII 字符

表对每个字符进行判断，将连续的可打印字符归为普通文本，而将连续的不可打印字符归为非文本。

4) 句子的解析：将连续的几个有意义单词合并为句子类型。利用一个常见单词字典作为有意义单词的判断标准。

为了后续操作的方便，将流的方向也作为一个 Token，图 3 是一个 Token 解析的例子，图中“\x00”表示从发起端到回应端的流，“\x01”则表示反方向的流。

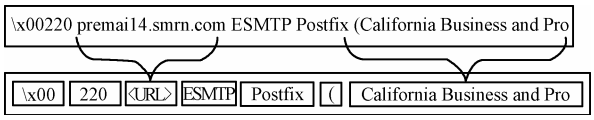

图 3　Token 解析过程

3.2.3　Token 序列聚类

以 Token 序列为基础，可以归类相似的流量。本节将介绍 TPCAD 所使用的序列比较、合并及距离的表示方法，最后说明 Token 序列聚类及合并的具体过程。

1) 序列的比较及合并

2 个序列相同是指对应位置上的 Token 相同。由于 Token 具有不同的属性、类型和值，在比较时可变属性的 Token 只要类型相同就认为它们相同；固定属性的 Token 只有值相同时才相同。合并时，同类型的 Token 会被合并成可变属性的该类型的 Token，而不同类型的 Token 则会被合并为可变属性的普通字符。

2) 距离的表示

本文的聚类是以语义的相似度为基础的，只有语义结构相似的序列才可以分到同一类中。为此限定相似的 2 个序列具有以下 2 个前提：

前提 1　序列的长度相同，即 2 个序列具有相同的 Token 数量；

前提 2　序列的属性分布相同，即 2 个序列对应位置上的 Token 属性相同。

直观地讲，在 2 个前提下可以使用 2 个 Token 序列中 Token 具有差异的位置数来表示 2 个序列的距离。令 x 表示 Token 序列，则 $d(x_i,x_j)$ 表示 2 个各序列具有差异的位置数。但是在实际实验中这样会加大 2 个序列的距离，造成过分类。于是本文将 2 个序列中 Token 相同的位置数也加入到距离函数中，表示为 $s(x_i,x_j)$。由于可变属性 Token 的相同条件比较弱，需要一个系数调节它对距离的影响，在实验中发现结果对于这个系数并不敏感，因此指定 0.49 作为调节系数。使用下标 C 表示固定属性的 Token，使用下标 V 表示可变属性的 Token。最后，距离函数可以表示为如下形式：

$$D(x_i,x_j)=d_C(x_i,x_j)-s_C(x_i,x_j)+d_V(x_i,x_j)-0.49s_V(x_i,x_j)$$

当距离大于 0 时，认为 2 个 Token 序列具有差距，应该被分到不同的类中。

3) 聚类算法

为了达到聚类的前提条件，需要先将所有的 Token 序列进行排序，将满足前提条件的序列分到相同段中。然后在每一段中执行聚类算法并合并每一个类的序列。

在进行排序时，首先按照序列的长度进行排序，使满足第一个前提条件的序列排在一起。对于长度相同的序列按照每一个 Token 的属性从左至右进行排序，这样就使满足 2 个前提条件的序列都排在一起。进一步地继续按照 Token 的类型和值进行进一步的排序，从而使相似的序列都排在一起。排序完成之后，按照 2 个前提条件将序列分成不同的段。

对于同一段内的序列执行聚类和合并操作。算法开始时，每一个序列都是一个单独的类。每一个类都与其他类进行比较，如果距离小于 0 且不匹配位置相同，则合并 2 个类。开始时不存在不匹配位置，当 2 个类合并时含有不相同 Token 的位置会成为不匹配位置，这个位置会被记录下来用于下次比较。算法循环的对段内的类进行比较，直到没有新的类合并发生为止。

聚类结束后需要将每一个类处理成一个新的 Token 序列，把新生成的 Token 序列叫作 Pattern。每个类中的匹配位置是不需要改变的，但是需要将不匹配位置的不同取值合并成一个新的 Token。图 4 是一个合并时的例子。

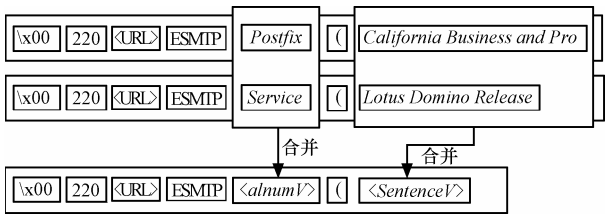

图 4　Token 序列聚类及合并过程

3.2.4　聚类关联

根据上面的聚类过程可以看出，TPCAD 聚类是很细粒度的，只有相似程度较高的序列才会被分为一类并合并到一起。采用细粒度的聚类策略是因为每个协议都会有几种不同的负载特征，每一个类都是一种负载特征。为了得到某一协议的特征集合，需要把几个 Pattern 关联在一起。

根据前面提到的，一个会话包含 2 条流，这 2 条流肯定属于同一个会话。以此为基础，可以得到四元组相同的 2 条流属于一个协议。进而可以得到四元组列表中包含相同四元组的 2 个 Pattern 属于相同协议。根据这一特性，本文设计了如下算法对 Pattern 进行关联。

首先，将所有 Pattern 按方向分为 2 组，并使用这 2 组 Pattern 构造一个二维矩阵。如图 5 所示，以“*P*0”开头的为从初始端到回应端方向的 Pattern，以“*P*1”开头的是反方向的 Pattern。方格中的数字表示横纵坐标的 2 个 Pattern 所对应的四元组列表中相同四元组的个数。方格中的数字大于 0 表示对应的 2 个 Pattern 应该关联在一起。遍历整个表格寻找可以关联在一起的 Pattern。

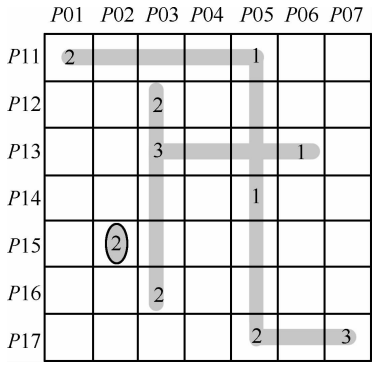

图 5　Pattern 关联算法

经过前面的处理之后，每一个协议都对应为一个 Pattern 集合，可以利用 Pattern 集合进行流量分类了。在实验部分本文也测试了直接利用 Pattern 集合进行流量分类的性能。

3.3　特征提取

Pattern 所表示的负载特征粒度较细，会造成每个协议都含有大量 Pattern，过多的 Pattern 会降低流量分类的速度。为此将同协议的 Pattern 进行合并，抽取它们共同的特征作为此协议最后的负载特征。首先将有可能合并的 Pattern 分为一组，然后对组内的 Pattern 进行对齐合并，最后抽取出公共的特征。

3.3.1　分组

根据对网络流量的观察，不同方向的流负载一般都带有不同的特征，即 Pattern 的第一个 Token 不同，其所代表的负载特征也不同。不仅如此，大部分协议在负载开头的区别都较小，而在负载的后部会出现较多的分支，因此第二个 Token 不同的 Pattern 也具有不同的负载特征。因此可以根据 Pattern 的前 2 个 Token 将 Pattern 划分为几个组。

为了后面组内对齐的方便，需要组内所有 Pattern 都具有至少一个相同的 Token，因此需要进一步的划分。在组内寻找出现次数最多的 Token，把含有此 Token 的 Pattern 分为一组。接着在剩下的 Pattern 中重复上面的操作，直到所有 Pattern 都被分到新的组中。为了排除常用词汇对分组结果的影响，可以忽略值为生活中常用单词的 Token。

为筛选出日常用词，需要建立一个判别标准，并据此进行筛选。本系统以语音识别领域常用的英文单词发音词典 Beep[12]为基础，稍加修改后作为日常用词的判别标准，凡是在词典中出现的字符串均属于日常用词。

3.3.2　对齐提取特征

在每一个 Pattern 组内至少含有一个出现在所有 Pattern 中的 Token，把这些 Token 叫作对齐点，如图 6 所示。对齐就是要找到一个最长的对齐点序列，这是一个最长公共子序列问题（longest common subsenquence）。对于输入仅为 2 个长为 N 的串的 LCS 问题，其复杂度是 $O(N^2)$，而对于输入串为 K 个长为 N 的串时，那么 LCS 问题的复杂度为 $O(K\times N^K)$。尽管这是一个高复杂度的算法，但是对于本文的问题而言并不能保证得到最优解。为此，本文设计了一个近似算法，并且对于本文的问题而言已经可以达到期望的效果。

在算法中随机选择一个 Pattern 作为基准 Pattern。基准 Pattern 中一定包含所有的对齐点，因此把其他 Pattern 与基准 Pattern 进行比较，记下每个 Token 出现的次数和出现在基准 Pattern 中的位置。处理完所有 Pattern 后，出现次数最多的 Token 便是对齐点，而且根据 Token 出现的位置可以得出最后的对齐点序列。例如，Pattern 组中有 3 个 Pattern：“ACDB”、“ABDC”和“ADBC”，本文选择 ACDB 作为基准 Pattern。在遍历一遍之后可以知道，出现过的所有 Token 都出现了 4 次，因此，A、B、C、D 都是对齐点，有可能出现在最终的结果里。

除基准 Pattern 外的 2 个 Pattern 的位置序列为“0 3 2 1”和“0 2 3 1”，找出出现在所有序列中的递增子序列“0 1”、“0 2”和“0 3”。最后选出最长的子序列所对应的 Token 序列作为最终对齐点序列，在本例中可以选择“0 1”，其对应的 Token 序列为“AC”。

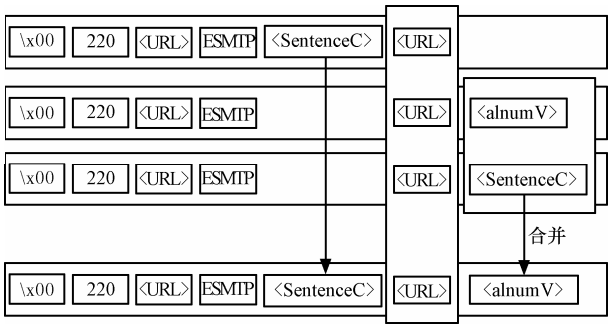

图 6　Pattern 对齐过程

对齐之后需要将非对齐点位置的 Token 进行合并，对于合并产生可变属性的 Token，保存其取值范围，这样可以尽可能地提高其准确程度。

4　方法验证

为了验证 TPCAD 方法在实际网络环境中的性能，使用实际流量对方法进行了验证。由于 TPCAD 方法和 Ma 的方法有类似的目的，本文也在实验中与 Ma 方法的结果进行了比较。

4.1　数据集

本文在国内某骨干网路由器上采集了 1h 的数据，流量文件中只包含了 TCP 流量。使用 L7-filter 对流量文件进行标注，通过预处理可知其中包含文本类协议的流 941502 条。为方便对比，本文选取 5 个常见协议对 Ma 方法和 TPCAD 方法进行测试，分别是 HTTP、SMTP、FTP、SSH 和 Gnutella 协议。将分类后的文本类协议数据平分为 2 组，一组用于训练，一组用于测试。

4.2　实验方法

为了进行比较，本文实现了 Ma 论文中的 2 种方法：乘积分布模型和马尔科夫分布模型。由于乘积分布模型比较适合特征出现在固定位置的情况，在变化比较复杂的广域网环境表现较差，本文没有把它的结果列在实验结果中。而马尔科夫过程模型比较适合文本类协议数据特征的描述，所以在实验中只与马尔科夫过程模型进行比较。

TPCAD 方法分为 2 部分：聚类过程和特征提取过程。聚类过程对输入向量进行解析并合并相似的 Token 序列，可以用来进行流量分类。分类时，现将流量中的数据按照算法中的步骤解析成 Token 序列，然后把这些 Token 序列与 Pattern 进行逐个比较，直到匹配为止。这样的分类方法在实验中叫作 Pattern 方法。Pattern 方法需要进行大量的解析和匹配操作，比较耗时。

而特征提取过程从 Pattern 中发现其中的共同点并提取出来，这样可以简化特征的表示方法，提高匹配速度，但是会引入一定程度误差。TPCAD 方法最后会把特征转化为正则表达式形式，然后把这些正则表达式加入 L7-filter，通过 L7-filter 来验证本文算法的性能。为了比较 2 个方案的差别，本文同时测试了 2 个方案的性能。

本文的评价指标包括机器学习领域常用的错误率、准确率和召回率，以及训练时间和识别匹配时间。首先介绍错误率、准确率和召回率的定义。

设测试数据总量为 N，如果将数据分为 A 与非 A 2 类，那么定义如下：

1) TP 表示 A 被正确识别为 A 的总量；

2) FN 表示 A 被错误识别为非 A 的总量；

3) FP 表示非 A 被错误识别为 A 的总量；

4) TN 表示非 A 被正确识别为非 A 的总量。

记正确率为 ACC，错误率为 ERR，准确率为 $PREC$，召回率为 REC，那么有：

$$ACC = \frac{TP + TN}{N}$$

$$ERR = 1 - ACC$$

$$PREC = \frac{TP}{TP + FP}$$

$$REC = \frac{TP}{TP + FN}$$

另外，训练时间指的是系统对训练数据开始进行训练到获得可用于网络流识别使用的协议特征为止所用的时间，而识别匹配时间指的是利用协议特征对所有测试数据进行识别匹配并获得识别结果所用的时间。

错误率越低，准确率和召回率越高，训练时间和识别匹配时间越少，相应的流识别方法越好。

4.3　实验结果

4.3.1　错误率

TPCAD 有着较低的错误率，与 Ma 方法相比也比较稳定。从图 7 中可以看出，Ma 方法在 HTTP

上有着较高的错误率，而在其他几个协议上有着不错的表现。这主要是因为 Ma 方法对简单协议有着比较好的效果，而 HTTP 是一个比较复杂的协议。HTTP 协议有着众多的可选参数，这使得其在流量中的表现形式多种多样。不仅如此，大量的其他协议将自己的流量隐藏在 HTTP 协议之中，或者使用 HTTP 的标准通信端口。这些都加剧了 HTTP 协议的复杂性，因此 Ma 方法所产生的转移矩阵会比较庞大和复杂。Ma 在文献[5]指出，复杂的转移矩阵会引入错误，因此 Ma 的方法在复杂协议上的表现不好。

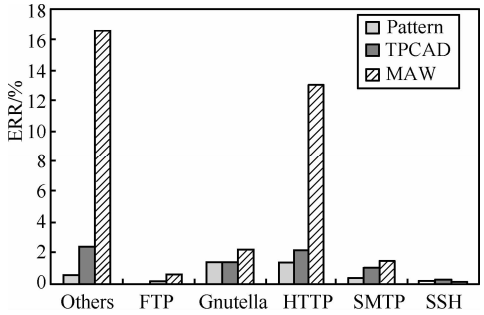

图 7　错误率对比

4.3.2　准确率

除去 Gnutella 协议外，Ma 方法和 TPCAD 一样都有着不错的表现。但是 Ma 方法不能识别 Gnutella 协议。这是因为 Gnutella 使用随机端口技术，而 Ma 的预分类标准是服务器 IP 和端口，所以 Ma 方法不能划分出 Gnutella 流量并提取特征，如图 8 所示。

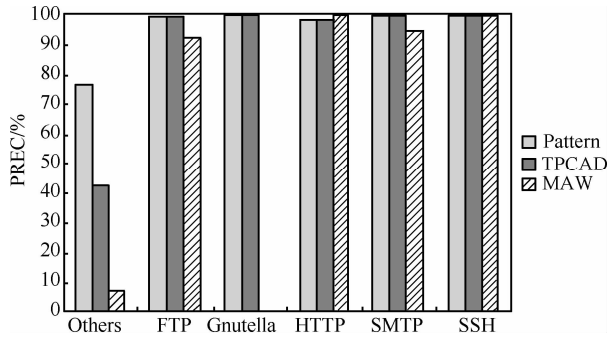

图 8　准确率对比

4.3.3　召回率

在召回率上，TPCAD 均好于 Ma 方法。但是在 Gnutella 协议上 TPCAD 仅有不到 40%的召回率。Gnutella 将自己的协议内容隐藏在 HTTP 协议之中，并且仅在一处出现明显的特征。因此在一个会话的 2 个流里只有一个流含有明显的特征，这对于所有流量分类方法来说都是一个挑战。由于互联网路由的特性，本文所采集的数据多数情况下仅含有会话中的一个流，因此本文方法的召回率很难超过 50%，如图 9 所示。

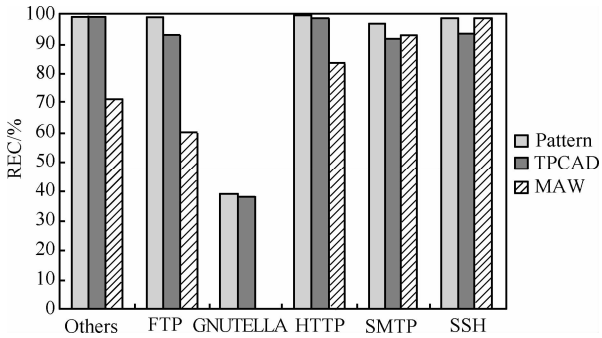

图 9　召回率对比

从前面的实验结果可以看出，Ma 方法对格式简单的协议效果比较好，在 SSH 协议上略微超出了 TPCAD 方法。但是它对复杂格式协议的效果较差，而对于采取隐藏措施的协议则无法分类。另外，Ma 方法对于其他类的协议效果较差，这显示出其对不同协议的适应能力较差。

4.3.4　时间消耗

在一台拥有 2.80GHz Pentium(R) D CPU 和 1G 内存的 PC 机上测试 TPCAD 和 Ma 方法，测试结果如图 10 所示。从图中可以看出，Ma 方法的训练过程十分耗时，这是由于马尔科夫过程模型的训练复杂度比较高。Pattern 方法和 TPCAD 方法速度都比较快，但是 Pattern 方法的分类时间要高于 TPCAD 方法，这和我们的预期是一样的。根据实验，TPCAD 方法的分类速度是 Pattern 方法的 7 倍，TPCAD 方法在牺牲少量准确程度的前提下，大幅提高了分类速度。

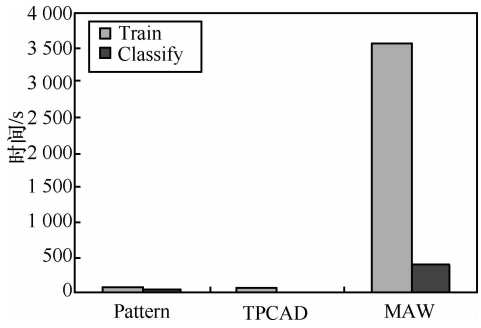

图 10　时间消耗对比

4.4　协议发现能力

Ma 的方法和 TPCAD 是少数的可以自动发现新协议的流量分类算法。因此本文专门比较了 2 个方法的协议发现能力。表 1 和表 2 是 2 个方法的协议发现

情况。表中的结果分为 2 部分，前面的整数表示流量被划分到多少个类中，后面的百分比表示被划分的流量所占的比例。从结果中可以看出，Ma 的方法把同一协议的流量划分到许多类中，并且对于复杂格式的协议类的数量很多。这是因为 Ma 采用服务器 IP 和端口作为预分类方法，而目前非标准端口的使用越来越多，这造成了 Ma 方法的预分类效果较差。TPCAD 采用语义相似性作为分类标准，效果比较理想。不仅如此，TPCAD 还发现了一些 L7-filter 不能识别的新协议，表 2 中的后 3 项是新发现的协议。

表 1　　Ma 的方法的协议发现能力

Product Distribution	HTTP	SMTP	SSH	FTP	MSN
	417/15%	26/20%	19/80%	12/65%	2/25%
Markov Process	HTTP	SMTP	SSH	FTP	MSN
	308/10%	26/10%	9/40%	8/60%	2/25%

表 2　　本文方法的协议发现能力

TPCAD	HTTP	RTSP	SMTP	GNUTELLA	SSH	FTP
	2/95%	1/95%	1/95%	3/40%	1/95%	2/95%
	IRC	MSN	SIP	TSPVIEW	P8NET	ASX
	2/95%	11/95%	1/95%	1	1	1

5　结束语

流量分类问题作为信息安全领域一个重要的研究方向已经得到越来越多人的关注。而基于负载特征的流分类方法由于准确率高而得到了广泛的应用。但是目前负载特征的获取主要依靠手工分析。本文利用网络流量中文本内容的语义特征，将流量解析成 Token。提出了一个在 Token 空间上的相似性比较方法，并据此对文本类的流量进行聚类。然后合并同一类的网络流量并提取出同类流量所共有的特征。实验表明，该方法对常见文本类协议拥有超过 95%的准确率，并可以实现对流量的在线学习和分类。

参考文献：

[1] IANA TCP and UDP port numbers [EB/OL]. http://www.iana.org/assignments/port-numbers, 2008.

[2] L7-FILTER. Application layer packet classifier for Linux[EB/OL]. http://l7-filter.sourceforge.net/, 2008.

[3] WIRESHARK. Network protocol analyzer [EB/OL]. http://www.wireshark. org/, 2008.

[4] CUI W, KANNAN J, WANG H J. Discoverer: automatic protocol reverse engineering from network traces[A]. Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium[C]. Boston, MA, USENIX Association. 2007.

[5] MA J, LEVCHENKO K, KREIBICH C, *et al*. Unexpected means of protocol inference[A]. Proceedings of the 6th ACM SIGCOMM conference on Internet Measurement[C]. Rio de Janeriro, Brazil,. 2006.

[6] CLAFFY K. Internet traffic characterization[D]. University of California, San Diego, 1994.

[7] ROUGHAN M, SEN S, SPATSCHECK O, *et al*. Class-of-service mapping for QoS: a statistical signature-based approach to IP traffic classification[A]. Proceedings of the 4th ACM SIGCOMM conference on Internet measurement[C]. Taormina, Sicily, Italy, 2004.

[8] MOORE A W, ZUEV D. Internet traffic classification using bayesian analysis techniques[A]. Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems[C]. Banff, Alberta, Canada, 2005.

[9] KARAGIANNIS T, PAPAGIANNAKI K, FALOUTSOS M. BLINC: multilevel traffic classification in the dark [J]. SIGCOMM Comput Commun Rev, 2005, 35(4): 29-40.

[10] HAFFNER P, SEN S, SPATSCHECK O, *et al*. ACAS: automated construction of application signatures[A]. Proceedings of the 2005 ACM SIGCOMM Workshop on Mining Network Data[C]. Philadelphia, Pennsylvania, USA, 2005.

[11] IETF. Uniform resource locators (URL)[EB/OL]. http://www.ietf. org/rfc/ rfc1738.txt, 2008.

[12] COMP.SPEECH. BEEP dictionary[EB/OL]. http://svr-www.eng.cam.ac.uk/comp.speech/Section1/Lexical/beep.html, 2008.

作者简介：

赵咏（1983-），男，河北石家庄人，中国科学院计算所博士生，主要研究方向为信息安全、流量分类、P2P 测量。

姚秋林（1983-），男，浙江庆元人，中国科学院计算所硕士生，主要研究方向为数据流查询、流量分类。

张志斌（1978-），男，山东济南人，中国科学院计算所助理研究员，主要研究方向为网络流处理、网络测量等。

郭莉（1969-），女，湖南株洲人，中国科学院研究员、高级工程师，主要研究方向为算法设计与分析、数据流管理、网络与信息安全。

方滨兴（1960-），男，江西万年人，博士，中国工程院院士，中国科学院研究员，主要研究方向为计算机网络及信息安全。