2011 年 5 月 第 5 期
Vol. 32 No. 5 2011

识别应用流量的一种新方法

王变琴[1,2],余顺争[1]

[1](中山大学 信息科学与技术学院,广东 广州 510006)

[2](中山大学 东校区教学实验中心,广东 广州 510006)

E-mail:wangbq@mail.sysu.edu.cn

摘 要:当今网络环境中,新型、未知应用大量涌现、并且网络技术日新月异,这对网络流量的识别带来严重挑战.传统的基于IANA端口的应用识别方法逐渐失效,利用流行为统计特征的流量分类方法在精度和实时性上又存在先天的缺陷.本文提出了一种新型的基于数据挖掘的应用识别方法,该方法从应用会话内容中自动提取应用特征,然后根据特征匹配识别应用.通过仿真实验测试识别率、正确率及综合指标,结果表明算法是有效性,能够实现应用层流量的精确分类.

关 键 词:流量识别;自动提取应用特征;频繁项挖掘;应用会话

中图分类号:TP393 **文献标识码**:A **文 章 编 号**:1000-1220(2011)05-0875-06

Novel Method for Identification of Application Protocol Traffic

WANG Bian-qin[1,2], YU Shun-zheng[1]

[1] (*School of Information Science and Technology, Sun Yat-sen University, Guangzhou* 510006, *China*)

[2] (*Education & Experiment Center, Sun Yat-sen University, Guangzhou* 510006, *China*)

Abstract: With the various newly designed applications and the rapid development of the network technology, the network traffic identification has encountered serious challenges. The traditional method of traffic identification based on the port number assigned by IANA could no longer be applied due to its deficiency, and another method by the statistical characters of flows could hardly satisfy the accuracy and real-time demand. This paper presents a novel approach to identify application traffic based on data mining, which automatically extracts application signature from the content of the application session, and furthermore identifies the same traffic using the signature matching. The identification rate, precision rate and F1-Measue have been verified on some traces and experimental results are presented to show the effectiveness of the algorithm, which can be used to accurately identify applications.

Key words: traffic identification; automatically extracted application signatures; frequent session fragment mining; application session

1 引 言

Internet、Intranet上运行有多种网络应用,准确地识别和区分它们的流量是提供差异性服务、QoS保障、入侵检测、流量监控、计费管理等应用的前提和基础.传统的应用识别采用端口(Port)识别法[1],这种方法能达到较高的速率,但大量的应用为了防止被识别和逃避防火墙的检测,不使用固定的端口进行通信,这不仅包括众多近年新出现的P2P应用(例如,BT),而且包括了越来越多的传统应用(例如,FTP)使用动态端口通信.此外,也存在复用公开端口进行通信(例如,HTTP,QQ应用则共用80端口).诸多的此类应用的产生,使得端口识别法已逐渐失效,因此,近年来越来越多的研究工作致力于探索识别应用层流量的新方法.

基于流(flows)行为特征的识别方法[2-5]运用统计学方法对应用流量的包长分布、流持续时间、平均包到达时间等数据进行分析,得到流量的统计分布特征.这种方法计算复杂度低、效率高.但统计特征存在一定的误差,识别的准确率不高,一般不能进行应用的精确判定,而且判定结果滞后,难以用其对流进行实时跟踪控制,因此,实际应用中较少采用.基于载荷(payload)特征串的识别方法[6-8]通过分析应用层协议,找出其交互过程中不同于其它应用的特征串作为该应用流量的识别特征,采用特征匹配的方式进行流量识别.这种方法准确识别的前提是正确找出应用特征,其完备性、准确性对识别率、准确率有极大的影响.

应用特征自动构成(Automated Construction of Application Signatures, ACAS)方法[9]将TCP数据流(data stream)的前64个字节作为特征向量.这种基于未经特征选择的方法的缺点是特征向量的数量大,训练时间长,同时基于Naive Bayes,AdaBoost,Maximum Entropy等机器学习的特征识别方法本身就存在一定的误差和不确定性,并且随着应用数量增多误差将迅速增大;利用反向工程(reverse engineering)的识别方法[10-13]为应用识别提出了新的途径,但现用方法需要了解应用协议报文格式的语法规范;基于正则表达式(regular expression)的应用识别[14](例如,Linux中的L7-filter[15])相对于其它方法其识别的正确性、效率有很大提高,但目前识别的应用数量有限,并且需要人工归纳应用特征来构造每种应用的正则表达式.

收稿日期:2010-03-10 收修改稿日期:2010-05-27 基金项目:国家自然科学基金-广东联合基金重点项目(U0735002)资助;国家"八六三"高技术研究发展计划项目(2007AA01Z449)资助;国家自然科学基金项目(60970146)资助. 作者简介:**王变琴**,女,1963年生,博士研究生,高级工程师,主要研究方向为网络应用识别、行为分析与控制;**余顺争**,男,1958年生,教授,博士生导师,主要研究方向为Internet流量测量、分析、建模,统计异常检测,无线网络.

网络应用(network application)种类繁多、实现过程复杂,不同应用有不同格式规范,使其特征提取变得非常困难.通过查阅应用层协议的有效文档(例如,HTTP 协议的标准文档 RFC2616)找到已知应用特征的方法对于不公开文档的应用(例如,MSN 应用)及不断出现的新应用是无能为力的.通过 wireshark[16]、tcpdump[17] 等捕包工具对网络上采集的应用层数据进行分析统计寻找应用特征的方法虽然能解决上述方法不能解决的问题,但其效率与可信度都较低,同时应用协议版本不断更新,新应用不断涌现的现实给这种半人工分析方法带来巨大的挑战.本文提出一种应用层流量识别的新方法,该方法以应用会话为研究对象,在特征提取和识别过程中充分考虑了应用层协议的结构特征,取得较好的应用识别效果.

2 相关概念与术语介绍

应用特征在会话中表现出的特性是特征提取算法设计的依据;从捕获的单一应用流量(Traces)中划分会话及对其数据进行重组是特征提取算法实现的前提和基础.

2.1 应用特征

定义 1. 应用特征(application signature,*AS*):在应用层数据中频繁出现的、并且具有位置特性的字节或字节组合.主要有两类:①应用层协议报头中的特征串,包括协议名称、版本号等,例如,HTTP 协议报文中的"HTTP/1."代表 HTTP 协议的名称及版本号,BitTorrent 协议报文中第 2 ~ 20 字节的值"BitTorrent Protocol"代表 BitTorrent 的协议名称;②应用层协议控制信息中的特征串,包括命令码、状态码及定界符等,例如,FTP 协议报文中的命令码"PASS"、响应码"220"以及回车换行符号"\0x0d0a"等.

在通信过程中应用特征具有如下特性:它们不一定出现在传递的每个报文中,但会频繁地出现在相应会话(定义 4)中;此外,应用特征还具有位置特性.一般情况下,特征出现在会话建立后的开始若干报文中,也有少数特征出现在会话结束的某个或几个报文中.特征在报文中的位置也有其特点:对于多数应用,特征出现在报文开始几个字节处(例如,SMTP 协议中的"HELO","250"等),但在某些应用中报文结尾的几个字节是结尾标志码(例如,QQ 协议报文的最后一个字节为"0x03").为了描述应用特征的这种位置特性,引入报文偏移量(定义 2)和字节偏移量(定义 3).

定义 2. 报文偏移量(message offset,*MO*):指示特征在会话中的报文位置(图 1 所示).若把会话建立后通信双方传递的第 i 个报文的 *MO* 标记 i,则其中的特征项的 *MO* 也为 i.

定义 3. 字节偏移量(byte offset, *BO*):指示特征在报文中的字节位置(图 1 所示).如果从报文开始处向结束方向标注报文中的字节,若把每个报文开始的第 k 个字节的 *BO* 标记为 k,则以该字节为首字节的特征项的 *BO* 也为 k.也可以按照相反方向进行标注,例如,若把每个报文倒数的第 l 个字节的 *BO* 标记为 l,则以该字节为尾字节的特征项的 *BO* 也为 l.

2.2 会话及其数据重组

根据 2.1 节的应用特征定义及其特性分析发现,在同种网络应用的流量中,相应的应用特征会相对频繁地出现在各个会话中,同时其在会话中也有其位置特点,这就需要应用特征提取要以会话报文序列为单位进行.然而实际流量数据是多用户、多种应用会话的无序混杂数据,因此,需要从捕获的训练 Trace 中提取完整的待测应用会话集,并且按时间顺序对其会话报文序列进行重组(reassembly).

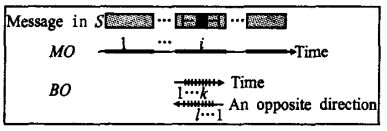

图 1 位置偏移量

Fig. 1 Position offset

定义 4. 会话(session, *S*):即一次通信建立和结束之间的所有报文构成的序列.假设每个会话包含至少 2 个报文:m_1, m_2, … 如图 2 所示,其中下标代表每个会话在时间上发生的顺序,奇数下标代表从会话的 A 端到 B 端传输的 messages,偶数下标代表从 B 端到 A 端的 messages.

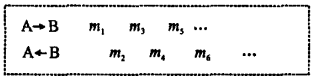

图 2 一个会话的报文序列组成

Fig. 2 A session composed of message sequence

数据重组方法:首先,收集单一应用的流量(Trace)作为该应用流量的训练样本集,然后,依据分组报头(header)的二元组(source IP、destination IP)信息及 TCP 数据流传输层会话建立和结束的标志(SYN, ACK, RST/FIN)将其划分成不同会话,并同时将每一个会话按照时间顺序依次保存其报文数据到一个数据文件中,完成其重组过程;对于 UDP 流量,其会话的开始、结束标志是空闲时间(idle time)大于一个给定阈值(常取 64s).在会话报文重组基础上,实现基于会话的应用特征自动提取.

3 应用识别

基于会话的应用识别首先需要从待测应用会话样本集中提取识别特征,然后实现基于特征的在线实时识别,本节内容主要介绍其基本框架及核心算法.

3.1 识别算法结构

应用识别模型由训练过程和识别过程组成,如图 3 所示.

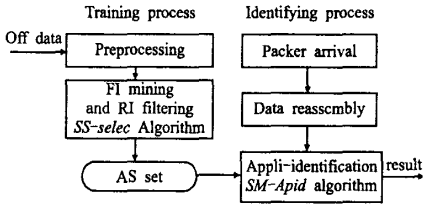

图 3 识别算法框架

Fig. 3 The framework of application identification algorithm

训练过程:训练过程包括数据预处理和特征提取(*SS-selec* 算

法). 数据预处理主要完成数据的重组,*SS-selec* 算法实现基于应用会话集的特征自动提取,在 3.2 节中将对其进行详细描述.

识别过程:在线识别过程包含应用层数据重组和应用识别(*SM-APid* 算法). 当报文到达后,按照时间顺序将同一个会话中的报文(message)内容存放到同一个缓存中,并将缓存的内容作为一段普通的文本,利用 *SM-APid* 算法(见 3.3 节)实现基于特征匹配的应用识别,其输出结果为分组所属会话的应用类型.

3.2 应用特征提取算法

应用特征提取就是从应用层数据中提取能够代表某种应用的全部特征的集合. 在通信过程中,应用特征一般具高频率(在一种会话集中反复出现)和关联性(在一个会话中同现),同时在一个会话中的偏移量(*MO* 和 *BO*)是相对固定的,据此,本文提出一种特征自动提取(signature set selection, *SS-selec*)算法,该算法是对经典的关联规则发现算法—Apriori 算法[19]进行了改进,使其适合于提取应用会话中的频繁会话片段集,再经过适当过滤规则筛选得到代表某种应用的特征集(signature set).

3.2.1 频繁会话片段挖掘

本文结合网络流量的特点提出基于会话的频繁会话片段挖掘(frequent session fragment mining based on session, *FSF-mining*)算法,其主要思想为:将待测应用的会话集视为交易数据库(transaction database),其中的会话(格式为 binary)视为关联事务(associative transaction),在给定的支持度(定义 5)阈值下,挖掘出待测应用会话中的频繁会话片段(frequent session fragment, *FSF*).

设 $I=\{I_1, I_2, \cdots, I_h\}$ 为项(item)的集合,其中 I_i 为单字节项. 对任意 X 满足 $X\subseteq I$,称 X 为一个会话片段集,如果 X 包含 k 个项(或 bytes),则称 X 为 k 会话片段集,记为 *k-itemset*. 设 $D=\{d_j\}$ 为 I 上的单一应用会话数据库,其中 $d_j(j=1,2,\cdots,n)$ 为第 j 个会话,并且 $I_{jl}\in I$,D 中共有 n 个会话.

定义 5. 支持度(support,记为 *sup*):给定 D 和 X,称 $sup(X)=n(X)/n$ 为 X 在 D 上的支持度,简记为 $sup(X)$,其中 n 为 D 中会话总数,$n(X)$ 为 D 中包含 X 的会话数.

定义 6. 频繁会话片段集(frequent session fragment set, *FSFs*):给定 D、X 及最小支持度 $min\_sup\in(0,1)$,当 $sup(X)\geq min\_sup$ 时称 X 为 D 上的频繁会话子集,D 上所有的频繁会话片段子集组成的整个集合称为频繁会话片段集,记为:$FSFs=\{X \mid X\subseteq I\wedge(sup(X)\geq min\_sup)\}$.

性质 1. 频繁会话片段集的任何子集也是频繁的. 例如,字符串(abc)∈3-*item* 意味着其子串(ab)∈2-*item*,(bc)∈2-*item*.

假设待测应用的会话集中有 n 个会话,最小支持度为 *min_sup*,利用 *FI-mining* 算法提取 *FSFs* 的原理步骤如下:

Step 1. 挖掘 1-*itemset*:计算每个 1-*item* 的 $sup(1\text{-}item)$,当 $sup(1\text{-}item)\geq min\_sup$ 时,将其列入 1-*itemset*;

Step 2. 利用 1-*itemset* 获取 2-*itemset*:即连续的两个字节,计算每个 2-*item* 的 $sup(2\text{-}item)$,当 $sup(2\text{-}item)\geq min\_sup$ 时,将其列入 2-*itemset*;

Step 3. 由 *k-itemset* 获取 $(k+1)$-*itemset* $(k\geq 2)$:对于 *k-itemset* 中的任意两项 l_1 和 l_2,如果满足连接条件,则将其合成一个 $(k+1)$-*item*,然后计算 $(k+1)$-*item* 的 $sup((k+1)\text{-}item)$,当 $sup((k+1)\text{-}item)\geq min\_sup$ 时,将其列入 $(k+1)$-*itemset*;如此进行,直到没有更长频繁字串为止,设最长频繁字串为 l;

Step 4. 生成 *FSFs*:将 1-*itemset*、2-*itemset*、3-*itemset*、…、*l-itemset* 合成为 *FSFs*,即 $FSFs=(1\text{-}itemset)\cup(2\text{-}itemset)\cup(3\text{-}itemset)\cup\cdots\cup(l\text{-}itemset)$.

要挖掘出最完备的可能特征,在初次挖掘 *FSF* 时,*min_sup* 阈值要尽可能小,可以取最小值,即 $min\_sup=1/n$. 但在实际应用中,为了提高算法的效率,在大多数情况下,*min_sup* 的初始值可以设为 0.5(通过实验获得的经验值),然后,根据结果进行上下调整.

3.2.2 冗余项过滤

FSF-mining 算法挖掘出的 *FSFs* 中会包含许多冗余项. 冗余项(redundant item, *RI*)是指在会话中频繁出现,但又不能作为应用特征的字符串,它们的存在会影响识别应用的准确性,因此,需要对其进行过滤.

根据性质 1,*FSF-mining* 算法会产生许多包含关系的 *FSF*. 设 $FSFs(x)(\subset FSFs)$ 是 $FSFs(y)(\subset FSFs)$ 的子串,则有 $sup(FSFs(x))\geq sup(FSFs(y))$,并且 $sup(FSFs(x))=sup(FSFs(y))$ 的情况属于大多数. 在一组包含关系的 *FSF* 中,往往只有一项可能是特征项,其余项都可以视为冗余项,例如,SMTP 应用流量的 *FSFs* 中,如果特征码"250"被选中,则其子串"2"、"5"、"0"、"25"及"50"也一定会被选中,对此设置过滤规则 1 对其进行过滤.

过滤规则 1. 若 $FSFs(x)(\subset FSFs)$ 是 $FSFs(y)(\subset FSFs)$ 的子串,并且 $sup(FSFs(x))=sup(FIS(y))$,即 $FSFs(y)$ 出现的次数与 $FSFs(x)$ 相等时,则认为所有的 $FSFs(y)$ 出现时都包含了 $FSFs(x)$,此时过滤 $FSFs(x)$.

这是一条强规则,它滤掉了包含关系中的绝大多数冗余项. 对于极少 $sup(FSFs(x))>sup(FSFs(y))$ 的情况,若保留 $FSFs(x)$ 可以获得较高的识别率,若保留 $FSFs(y)$ 可以保证较低的误报率,因此,两者均给予保留.

此外,也存在由于一些应用协议的请求/响应报文本身携带实体数据(例如,HTTP 协议),或由流量中的其它报文数据引起的高频数据项,这类冗余与应用特征的根本区别在于频繁特征在会话中出现的位置是较为固定的,而冗余的分布则一般是随机的. 因此,在原算法的基础上增加对 *FSF* 的位置标记,通过记录每个 *FSF* 在报文中的 *BO*,确定其分布的随机程度,据此对特征和冗余进行区分.

过滤规则 2. 扫描频繁项 $FSFs(i)$ 在每个报文中的 *BO*,计算其位置自由度(定义 7),通过设置其阈值(threshold)(严格地讲,$pos\_fre(FSFs(i))=1/n\_meg(FSFs(i))$,即固定偏移)对其进一步过滤.

定义 7. 位置自由度(position freedom,记为 *pos_fre*):描述 $FSFs(i)$ 在报文中出现的位置的随机性. 假设包含 $FSFs(i)$

的报文数量为 $n\_meg(FSFs(i))$,$FSFs(i)$在报文中出现的不同偏移位置(BO)数为 $n\_BO(FSFs(i))$,则 $FIS(i)$的位置自由度 $pos_fre(FSFs(i))$计算公式为:

$$pos\_fre(FSFs(i)) = \frac{n\_BO(FSFs(i))}{n\_meg(FSFs(i))}, \frac{1}{n\_meg(FSFs(i))}$$

$$\leqslant pos\_fre(FSFS(i)) \leqslant 1 \quad (1)$$

根据特征项在报文中具有相对固定的 BO 这一特性,$pos_fre(FSFs(i))$越大,说明 $FSFs(i)$出现的位置随机性越大,其不是特征项的可能性越大.

特征项作为应用的标识,一般要求其具有应用唯一性,即仅在待测应用中出现.这就需要过滤具有负支持度(negative support)(即在负例子集中的支持度)的 FSF,因此设置了过滤规则3,以尽量保证应用特征的唯一性.

过滤规则3. 检查频繁项 $FSFs(i)$($\subset FSFs$)在负例子集中是否出现,一旦出现,则将其从 $FSFs$ 中删除.

负例子(negative example)的选取对过滤规则3的过滤结果影响较大,通常是与待测应用相近的或者容易混淆的应用.在实验中通常将训练数据集中待测应用之外的其他应用流量作为负例子集,可能不能100%的保证应用特征的唯一性.对于漏网的非唯一特征项在混合应用流量测试阶段可以根据识别率和准确率的反馈进行排查消除.

经过以上规则的过滤得到不含冗余或冗余最少的应用特征集(signature set),用于待测应用的识别.

3.3 应用识别算法

基于应用特征建立一种简单的应用识别(signature match-based application identification, *SM-APid*)算法,它可以高精度实时识别连接会话所属的应用. *SM-APid* 算法的原理步骤如下:

Step 1. 计算特征匹配度:一般情况下,应用特征在一个会话中的偏移量(MO 和 BO)是相对固定的,据此提出如下的匹配度计算方法:

设 $C=\{c_1,c_2,\cdots,c_m\}$ 为应用类型集合,其中 m 是类别数量,c_i 是类别,$PS_i=\{ps_{i1},ps_{i2},\cdots,ps_{in}\}$ 为应用 $c_i \in C$ 的特征集合,若特征 $ps_{il} \in PS_i$ 出现在 c_i 会话建立后的第 j 个报文中,并且特征项的首字节出现在该报文的第 k 个字节处,则标注 ps_{il} 为 $ps_{il}(j,k)$,其中 j 为 ps_{il} 的 MO,k 为 ps_{il} 的 BO.

在流量识别时,设缓存区数据 Ms 是会话建立后的第 l 个报文,其报文的 BO 为 h,当且仅当$(j=l) \wedge (k=h) \wedge (ps_{il}(j,k) \in Ms)$时,则认为报文 Ms 匹配 ps_{il} 成功;若$(j=k) \wedge (k=h) \wedge (ps_{il}(j,k) \notin Ms)$,则认为 Ms 匹配 ps_{il} 失败.对于一个会话,设特征集合 PS_i 中匹配成功的特征数为 n_machi,匹配失败的特征数为 n_umachi,则定义特征匹配度 $sim(c_i)$:

$$sim(c_i) = \frac{n\_mach_i}{n\_mach_i + n\_umach_i} \quad 0 \leqslant sim(c_i) \leqslant 1 \quad (2)$$

Step 2. 会话到应用的映射:设 $sim=\{sim(c_i) \mid c_i \in C\}$ 为匹配度集,若 $Sim(c_m)=\max(sim)$,则识别该会话为应用 c_m 的数据流也可以表示为:

$$c_m = \arg\max_m sim \quad (3)$$

即将会话映射到与其具有最大匹配度的特征集所代表的应用.当 $\max(sim)=0$ 时,则判定数据流不能识别,可能为未知或新应用.

SM-APid 识别方法属于报文一次性匹配方法,即存储一个数据流已经到达的报文,直到其报文到达一定数目,再一起送去匹配,如匹配,则后续报文无需再送去匹配,否则表示这个数据流无法识别.这种匹配方式的最大优点是有利于跨报文的匹配,同时匹配速度快,正确率高.报文数量的选择是 *SM-APid* 匹配方法的关键,它直接影响识别效果.理论上,报文数量 N 越大,识别结果越精确,但 N 越大,识别所需的时间越长.实验结果发现,对于大多数应用,会话的前4个报文(不包含建立会话时的协商报文)中的特征足以识别协议.

4 实验评估

选择了目前较为流行的7种应用(HTTP, FTP, SMTP, POP, MSN, BT and SSL,它们分别代表网页浏览类、数据传输类、邮件收发类、即时通信类、P2P下载类和加密类)流量进行测试.实验主要包括两部分:特征提取和基于特征匹配的应用识别效果测试.实验环境为一台个人PC机(CPU:Intel Core 2 Duo E6550 2.33 GHz,内存:0.99GB),其测试工具为 matlab7.1.

4.1 数据及其预处理

测试数据(表1)主要源于中国教育与科研网(CERNET)某小区的网络出入口处的真实流量,有少量训练数据(FTP应用)是在实验室中仿造的模拟流量.数据均用 Wireshark[17] 采集,格式为 PCAP,数据内容包括头部信息(header)和载荷(payload).

表1 训练数据集和测试数据集

Table 1 Training data and test dataset

Application	Training Dataset		Test Dataset	
	Size(Mb)	*S(Total)*	*Size(Mb)*	*S(Total)*
HTTP	19.30	220	17.76	250
FTP	20.53	66	25.56	230
SMTP	15.08	163	10.10	154
POP3	18.20	128	16.80	110
MSN	04.63	139	03.06	211
BT	14.90	395	13.60	382
SSL	13.08	200	10.72	197

特征提取的结果与训练数据关系密切,为了尽可能获得全面代表各种测试应用的完备特征集,从两方面保证训练样本数据集的完备性:

(1) 利用 Wireshark 及正则表达式(regular expression)对混合数据流 Trace 中提取的待测应用流量进行了进一步验证及杂质过滤,以尽量保证训练数据集的正确性和纯净性;

(2) 剔除了训练数据集中的不完整会话,使其包含的每个会话尽可能完整,即包含一次连接的两个主机交互过程的完整报文序列.

利用2.2节中介绍的方法对数据进行重组处理,其输出结果用于待测应用特征提取.在特征提取时,每次输入纯净的单一应用的离线数据,挖掘出该应用特征.提取的应用特征集

的完备性、可靠性一定要根据多应用识别性能指标判定,同时也可以根据各指标值的优劣对特征集进行进一步优化.

4.2 应用识别测试

为了确保识别结果的可信性,测试数据全部来源于上述真实网络流量,采集时间为 2008 年 3 月. 由于应用层数据的数据量较大,考虑到测试机的内存容量,每种应用仅仅选取适量的会话(表 1).

识别测试是基于会话进行的,为此定义了 3 个性能度量指标:识别率(Identification rate,*IR*),正确率(Precision rate,*PR*)及两者的综合评价指标 F_1-Measue(简称 F_1). 其中识别率指示识别的完备程度,即应该正确识别的会话中有多少会话被正确识别,其大小可以反映特征集的完备性,其计算见公式(4). 正确率指示识别结果中有多少是正确的,其大小可以反映测试应用特征的区分度好坏,其计算见公式(5). 综合指标 F_1 指示两者的综合考察效果,可以综合反映所提取特征集的完备性、准确性及可靠性,其计算见公式(6). *IR*、*PR* 及 F_1 的计算公式如下:

$$IR = TP/(TP + FP) \tag{4}$$

$$PR = TP/(TP + FN) \tag{5}$$

$$F_1 = (2 \times IR \times PR)/(IR + PR) \tag{6}$$

其中,*TP*(True positive)表示将 c_i 应用的会话识别为 c_i 应用的数量,*FN*(False negative)表示将 c_i 应用的会话识别为非 c_i 应用的数量,*FP*(False Positive)表示将非 c_i 应用的会话识别为 c_i 应用的数量.

由于网络流量识别是在线实时环境,缓冲区读进的报文数量会随着时间的增长而增加,因此,测试过程模拟了读取不同报文数量的识别效果,据此可以确定精确识别会话所需要的报文数. 实验结果表明,当会话中的报文偏移 $N=4$ 时、并且取每个报文的前 20 个字节时,各种测试应用的度量指标几乎都达到最佳(见表 2、表 3),之后不再有明显的变化.

表 2　流量识别结果($N=1$,$N=2$)

Table 2　Traffic identification results($N=1$,$N=2$)

Application	$N=1$ Time consuming: 0.797s			$N=4$ Time consuming: 0.922s		
	IR%	*PR*%	F_1%	*IR*%	*PR*%	F_1%
HTTP	100	98.59	99.29	100	98.59	99.29
FTP	0	N/A	N/A	60	100	75
SMTP	81.08	98.36	88.89	90.54	73.86	80.25
POP3	95.65	98.51	97.06	97.10	97.10	97.10
MSN	97.69	100	98.83	97.69	100	98.83
BT	100	100	100	100	100	100
SSL	84.50	100	91.60	93.02	100	96.39

注:N 代表取会话开始的前几个报文

识别率 *IR*:当 $N=1$ 时,HTTP、POP3、MSN 及 BT 应用的识别率都达到 95% 以上,说明这几种应用的特征多集中在会话的第 1 个报文;由于交叉项的消除,FTP 的第 1 个报文的高频特征"220"不再用于识别,使其在 $N=1$ 时的识别率为零. 但当继续读取后面的报文时,FTP 应用的其它特征得到匹配,当读取到第 3 个报文时其能识别率已经接近 100%,以后再增加读取包数识别率也没有明显增长;SMTP、SSL 应用的识别率则随着 N 的取值增加而相继达到 94% 以上,表明其特征分散在会话的多个报文中. 在 $N=4$ 时,各种应用的高识别率(94.59% ~ 100%)表明获得的特征集是较为完备的,同时它们多集中在会话的前 4 个报文. SSL 应用的高识别率说明这种基于字节串的频繁项挖掘方法对类似 SSL 的加密应用同样有效.

表 3　流量识别结果($N=3$,$N=4$)

Table 3　Traffic identification results($N=3$,$N=4$)

Application	$N=3$ Time consuming: 1.156s			$N=4$ Time onsuming:1 .438s		
	IR%	*PR*%	F_1%	*IR*%	*PR*%	F_1%
HTTP	100	100	100	100	100	100
FTP	99.57	100	99.78	99.57	100	99.78
SMTP	93.24	97.10	94.59	94.59	98.55	95.10
POP3	98.55	97.14	97.84	98.55	97.14	97.84
MSN	99.53	100	99.76	100	100	100
BT	100	100	100	100	100	100
SSL	92.24	100	95.96	94.57	100	97.21

注:N 代表取会话开始的前几个报文

正确率 *PR*:除了 $N=2$ 时,SMTP 有较低的正确率(73.86%)外,其它应用在各种情况下的正确率都在 97% 以上(即误报率 <3%),表明所提取的特征的大部分区分度(独特性)很好. 在 $N=2$ 时,SMTP 的低正确率是因为其报文偏移量为 2 的特征在 FTP 应用流量中也存在,但不是频繁特征,因此 FTP 的特征集中没有包含该特征,由此造成大量 FTP 流量误报为 SMTP 流量;当 $N=3$ 时,随着其他报文特征的匹配,FTP 达到 100% 的识别,SMTP 应用的正确率达到 97.10%,它们得到很好的区分. 这表明要区分 SMTP 和 FTP 流量,N 的取值不能小于 3. 一般情况下,正确率的高低指示特征集的区分度大小. 产生高误报率的特征集,可能存在区分度较差的特征或特征冗余项,可以根据实际情况对其进行取舍.

综合指标 F_1:整体上看,在 $N=3$ 和 $N=4$ 时,多数测试应用的 F_1 明显高于 $N=1$ 和 $N=2$ 时的相应值,表明随着 N 取值的增加,匹配的特征数的增加,使多数应用的识别率和正确率提高,从而改善了指标 F_1. 在 $N=4$ 时,测试应用的值都达到 95% 以上,说明多数应用在会话的前 4 个报文中就可以提取到较为完备、准确的应用特征.

从实验结果分析可以看出,本文给出的算法是有效的、普适的,这种利用应用特征的流量识别方法与基于 flows 特征的识别方法[2-5]相比其识别粒度细、精确度高,可用于应用层流量的精确识别.

4.3 算法效率评估

应用识别算法除了高准确性外,还必须具有高效率以满足高速网络实时处理要求. 基于流量的特征自动提取算法—*SS-selec*算法与基于标准文档的手工或半手工的特征串提取方法[6-8]相比,极大地提高了特征提取的效率. 基于特征匹配度的 *SM-APid* 算法对测试的 7 种应用的混合流量(近 100M)的识别时间平均在 1s 左右,可见识别效率也较高,并且仅依据

应用流前期报文的若干字节就可以精确识别应用类型,可以满足高速链路的实时性处理要求,有利于流量的控制与跟踪.

实验发现,不同应用流量其特征的频繁程度差异很大,例如,对于 HTTP 协议,当 *min_sup* = 0.72 时可以获得完备的特征集,而对于 MSN 应用,当 *min_sup* = 0.51 时才可以获得完备的特征集,因此采用相同支持度阈值显然不合理. *SS-selec* 算法与刘兴彬等人[18]提出的方法相比,对不同应用采用不同支持度阈值,这样防止了相同支持度情况下某些应用特征集中可能包含大量冗余项,而另一些应用的特征集却不够完备,同时过滤算法简单、开销较小,可操作性强,因此算法的性能及实用性增强.

5 结束语

本文研究了应用流量识别问题,提出了基于数据挖掘的应用特征提取—*SS-selec* 算法和基于特征匹配度的应用识别—*SM-APid* 算法,并通过实验验证了方法的可行性和有效性. 这种方法的最大优势表现在它能够自动提取应用特征,既可以应用于已知应用,也可以应用于未知应用,并且由于基于少量特征的简单字串匹配,它的计算量相对较小,能够实现早期应用识别(测试的 7 种应用,在会话的第 4 个报文实现应用精确识别),同时这种方法不需要应用协议规范的先验知识,因此,当协议规范的细节发生变化或者新协议出现时,可以基于流量在线或离线更新特征集. 这种方法对于网络动态变化(例如,网络拥塞)及常见的路由不对成,丢包等网络现象不敏感,因此方法具有良好的健壮性和稳定性. 然而,作为一种新方法提出,仍有许多问题值得进一步研究:

(1) 对于不同应用,*SS-selec* 算法的 *min_sup* 参数的合理设置需要经过多次测试调整方能得到最佳值,实现此参数的自适应调整及算法的性能优化是进一步要解决的问题;

(2) *SM-APid* 算法虽然有较高的识别精度,但随着应用数量的增加、特征集的扩大,识别效率会下降. 因此,探索高效、高精度的特征匹配方法(例如,基于正则表达式的特征匹配算法)将是今后的另一个研究方向.

References:

[1] Internet assigned numbers authority (IANA) [EB/OL]. http://www.iana.org/assignments/port-numbers, 2007.

[2] Karagiannis T, Papagiannaki K, Faloutsos M. BLINK: multilevel traffic classification in the dark[C]. Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, New York, NY: ACM, 2005, 35 (4): 229-240.

[3] Zander S, Nguyen T, Armitage G. Automated traffic classification and application identification using machine learning[C]. Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05), Piscataway, NJ: IEEE, 2005, 250-257.

[4] Crotti M, Dusi M, Gringoli F, et al. Traffic classification through simple statistical fingerprinting[J]. ACM SIGCOMM Computer Communication Review, 2007, 37(1): 1-16.

[5] Bernaille L, Teixeira R, Akodkenou I, et al. Traffic classification on the fly[C]. ACM SIGCOMM Computer Communication Review, 2006, 36(2): 23-26.

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

[7] Moor A, Papagiannaki K. Toward the accurate identification of network applications[G]. LNCS 2172: Passive and Active Network Measurement, Berlin: Springer, 2005, 41-54.

[8] Choi Yongmin. On the accuracy of signature-based traffic identification technique in IP networks[C]. Proc of the 2nd IEEE/IFIP International Workshop. Piscataway, NJ, IEEE, 2007, 1-12.

[9] Haffner P, Sen S, Spatscheck O, et al. ACAS: automated construction of application signatures[C]. Proc of the 2005 ACM SIGCOMM Workshop on Mining Network Data, New York, NY: ACM, 2005, 167-202.

[10] Caballero J, Yin Heng, Liang Zhen-hai, et al. Polyglot: automatic extraction of protocol message format using dynamic binary analysis[C]. Proc of the 14th ACM Conference on Computer and Communications Security, New York, NY, ACM, 2007, 317-329.

[11] Cui Wei-dong, Kannan J, Wang H J. Discoverer: automatic protocol description generation from network traces[C]. Proc of 16th USENIX Security Symposium on USENIX Security Symposium, Berkeley, CA: USENIX Association, 2007, 14.

[12] Ma J, Levchenko K, Kreibich C, et al. Unexpected means of protocol inference[C]. Proc of the 6th ACM SIGCOMM Conference on Internet Measurement, New York, NY: ACM, 2006, 313-326.

[13] Shevertalov M, Mancoridis S. A reverse engineering tool for extracting protocols of networked applications[C]. Proc of 14th Working Conference on Reverse Engineering, Piscataway, NJ: IEEE, 2007, 229-238.

[14] Fan Hui-ping, Xuan Lei, Chen Shu-hui, et al. Speed up on application protocol recognition using regular express[J]. Journal of Computer Research and Development, 2008, 45(Sup.): 438-443.

[15] Application layer packet classifier for Linux[EB/OL]. http://17-filter. Sourceforge.net, 2009-05-01.

[16] Network protocol analyzer(wireshark)[EB/OL]. http://www.wireshark.org, 2009.

[17] Tcpdump[OL]. http://www.tcpdump.org/, 2009.

[18] Liu Xing-bin, Yang Jian-hua, Xie Gao-gang, et al. Automated mining of packet signatures for traffic identification at application layer with apriori algorithm[J]. Journal on Communications, 2008, 29(12): 51-59.

[19] Agrawal R, Srikant R. Fast algorithms for mining association rules in large databases[A]. Proceedings of the 20th International Conference on Very Large Data Bases, San Francisco[C], CA: Morgan Kaufmann Publishers Inc, 1994, 487-499.

附中文参考文献:

[14] 范慧萍,宣 蕾,陈曙晖,等. 基于正则表达式的应用层协议识别加速[J]. 计算机研究与发展,2008,45(Sup.):438-443.

[18] 刘兴彬,杨建华,谢高岗,等. 基于 Apriori 算法的流量识别特征自动提取方法[J]. 通信学报,2008,29(12):51-59.