doi:10.11959/j.issn.1000-436x.2015171

基于抽样分组长度分布的加密流量应用识别

高长喜[1,2]，吴亚飚[2]，王枞[1]
(1. 北京邮电大学 博士后流动站，北京 100876；2. 北京天融信公司 企业博士后工作站，北京 100085)

摘 要：基于确定性抽样数据分组序列的位置、方向、分组长度和连续性、有序性等流统计特征和典型的分组长度统计签名，并结合带数据分组位置、方向约束和半流关联动作的提升型 DPI，提出了一种基于假设检验的加密流量应用识别统计决策模型，包括分组长度统计签名决策模型和 DFI 决策模型，并给出了相应的分组长度统计签名匹配算法以及基于 DPI 和 DFI 混合方法的加密流量应用识别算法。实验结果表明，该方法能够成功捕获加密应用在流坐标空间中独特的统计流量行为，并同时具有极高的加密识别精确率、召回率、总体准确率和极低的加密识别误报率、总体误报率。

关键词：加密流量分类；应用识别；深度分组检测；动态流检测；混合方法
中图分类号：TP393.08 **文献标识码**：A

Encrypted traffic classification based on packet length distribution of sampling sequence

GAO Chang-xi[1,2], WU Ya-biao[2], WANG Cong[1]
(1. Postdoctoral Research Station, Beijing University of Posts and Telecommunications, Beijing 100876, China;
2. Enterprise Postdoctoral Working Station, Beijing TopSec Co., Beijing 100085, China)

Abstract: A hypothesis testing-based statistical decision model (HTSDM) for application identification of encrypted traffic was presented. HTSDM was based on packet length distribution of deterministic sampling sequence at flow level, which was characterized by packet positions, packet directions, packet sizes, packet arrival continuity and packet arrival order. HTSDM boosted deep packet inspection (DPI) by introducing constraints of packet position and direction as well as inter-flow correlation action. A hybrid method of encrypted traffic classification combining DPI and dynamic flow inspection (DFI) was proposed based on HTSDM. Experiment results show that this method can effectively identify the unique statistical traffic behavior of encrypted application in flow coordinate space, and achieve high precision, recall and overall accuracy while keeping low false positive rate (FPR) and overall FPR.

Key words: encrypted traffic classification; application identification; deep packet inspection; dynamic flow inspection; hybrid method

1 引言

网络流量应用协议识别是内容过滤、QoS、流量分析、安全通信及互联网监管和运维的基础。在网络安全领域，网络流量主要可分为明文流量、加密流量、匿名通信流量、入侵/攻击/渗透流量、病毒/木马/蠕虫/僵尸网络异常流量等。下一代网络中流量组成的复杂性及流量行为的多样性，特别是流量加密、伪装、隧道透传和分片等流量特征隐藏技术使网络流量应用协议识别面临着严峻的挑战。

根据所采用的协议特征的不同，应用协议识

收稿日期：2014-08-22；**修回日期**：2014-12-19
基金项目：中关村科技园区海淀园企业博士后工作专项基金资助项目（2012RC）；北京市博士后科研活动经费基金资助项目（2013ZZ-54）
Foundation Items: Enterprise Postdoctoral Research Support Program of Zhongguancun Haidian Science Park (2012RC)；Beijing Municipal Postdoctoral Research Support Program (2013ZZ-54)

别方法可分为：基于端口、深度分组检测（DPI, deep packet inspection）和动态流检测（DFI, dynamic flow inspection）[1]等。基于端口的应用协议识别方法将知名端口作为协议特征，例如 P2P 应用的固定服务端口、DNS 的 53 号端口等，然而动态端口、端口复用等机制使该方法已不能对应用流量进行精确分类。DPI 将数据分组载荷内部位置固定或变动的静态字节序列作为协议特征，或者通过深入可识别的信令通道提取协商数据通道的 IP 地址和端口而间接识别无特征数据流的应用协议类型（例如 SIP），支持数据流中的单个数据分组或多个数据分组协议特征，并可实现细粒度应用协议区分；然而随着网络应用（如 BitTorrent、eMule、Skype、Thunder 和 Tor 等）采用消息流加密或协议混淆来实现保密通信，除了极少数应用可通过逆向算法实时解密获取明文关键字特征之外，DPI 已无法有效识别加密类应用协议。DFI 将传输层连接模式、流统计特性[2]等流量行为作为协议特征，并使用启发式算法或机器学习方法进行流量分类，既能进行粗粒度应用分类（例如 P2P 类、交互类等），又能进行细粒度协议识别并且不依赖数据分组载荷内容（例如 SSH、HTTPS），因此该方法非常适用于加密流量应用协议识别。

本文基于 DPI 和 DFI 混合方法，提出了一种基于假设检验的加密流量应用识别统计决策模型，并给出相应的加密流量应用识别算法。该方法首次将确定性抽样数据分组序列的位置、方向、分组长度和连续性、有序性等流统计特征作为协议特征，给出了典型的分组长度统计签名，并通过单数据分组的位置、方向约束及半流关联动作提升了传统 DPI 方法。基于加密应用 BitTorrent 和 eMule 评估数据集的实验验证了该加密流量应用识别算法的有效性。

2 相关工作

当前加密流量应用识别的研究主要采用流统计特征的 DFI 方法。文献[3]基于流的指定方向上的前若干个数据分组的分组长度以及数据分组载荷前若干字节内容的概率分布定义了 34 种用于度量加密应用协议行为的统计属性指纹，提出了基于 K-L 散度（kullback-leibler divergence）的协议识别模型和算法，并通过实验评估了该算法识别混淆/加密协议的有效性；然而，该方法依赖载荷内容并且未充分利用流之间的相关性。文献[4]将流的支撑数据分组集合的分组长度分布作为协议特征，并根据端口局部性启发将流分组为会话，进而提出了一种基于距离相似性测度的会话级流分类方法，评估结果表明该方法对于流和会话都可以实现高准确率的分类；不过，该方法没有考虑到数据分组在流中的方向性。文献[5]基于流的前若干个数据分组在指定方向上的分组个数、分组长度及其均值、方差等给出了 17 种流量统计特征参数，提出了 *k*-means 和 *k*-nearest neighbors 机器学习算法相结合的加密流量混合分类算法，并在嵌入式实时环境中验证了该算法实时分类加密流量的可行性；但是，该方法未能将 DPI 与统计方法有效结合起来实现多重识别。文献[6]将流在指定方向上的分组长度与交互到达时间的最大值、最小值、均值、标准差和分组个数等作为流特征，并基于采集自不同网络的数据集评估了 AdaBoost、支持向量机（SVM）、Naïve Bayesian、RIPPER 和 C4.5 等 5 种机器学习算法用于分类 SSH 和 Skype 加密流量的顽健性，实验结果表明 C4.5 算法具有最优的分类性能。文献[7]将流的前若干个数据分组的带有方向标记的分组长度（经过缩放预处理）作为协议特征，基于 Gaussian mixture model 和 SVM 分类器对 SSH 隧道承载的应用协议进行识别，并通过对经过 SSH 加密的 POP3、POP3S、HTTP 和 eMule 的实验验证了该方法的有效性。文献[8]对近年运用机器学习方法进行 IP 流量分类的研究进展进行了综述和评论，将分类方法分为聚类、有监督的学习和混合方法等 3 类，并总结和比较了相关研究工作采用的具体机器学习算法、统计特征、评估数据集、待分类流量类型、分类粒度等策略以及准确率、实时性、计算复杂度和流方向依赖性等分类性能。

3 加密流量应用识别模型

基于假设检验的加密流量应用识别统计决策模型 HTSDM (hypothesis testing-based statistical decision model) 定义如下。

定义 1 流方向

流方向定义为由五元组（源 IP 地址、源端口、目的 IP 地址、目的端口和传输协议号）标识的流的数据分组发送方向，记为 $D_F=\{d_u,d_d,d_b\}$。其中，d_u 表示客户端向服务器发送分组的上行流方向，d_d 表

示服务器向客户端发送分组的下行流方向，d_b 表示不区分上下行的双向流方向。

定义 2　分组序列位置

分组序列位置定义为某个流方向上的带有有效负载的抽样数据分组序列的位置编号，并且位置编号在指定的流方向意义上针对全部带有有效负载的数据分组独立进行，记为 $X=\{x_i,\cdots,x_j|1\leqslant x_k \leqslant N,1\leqslant i\leqslant k\leqslant j\leqslant N\}$。其中，$x_k$ 表示单个数据分组的位置编号，称作分组位置；N 表示所在流方向上可取的最大位置编号。

根据所取的抽样位置序列$\{x_i,\cdots,x_j\}$的不同，分组序列位置可分为单个固定位置、离散序列位置和连续区间位置。所谓的离散序列位置是指具有不等长间隔的抽样位置序列，而连续区间位置是指步长固定为 1 的均匀位置序列。

定义 3　分组序列方向特征

分组序列方向特征定义为带有有效负载的抽样数据分组序列在流中出现时所位于的流方向，记为 $D_P=\{d_i,\cdots,d_j|d_k\in D_F,1\leqslant i\leqslant k\leqslant j\leqslant N\}$。其中，$d_k$ 表示分组位置为 k 的数据分组位于的流方向，称作分组方向；N 表示可取的最大分组位置。

定义 4　分组序列分组长度特征

分组序列分组长度特征定义为带有有效负载的抽样数据分组序列在流中指定的分组位置上的分组长度（即分组载荷长度）、分组长度序列、分组长度集合或分组长度统计量所应满足的特定阈值约束，记为 $L=\{l_i,\cdots,l_j|l_k=[inf_k, sup_k], i\leqslant k\leqslant j\}$。其中，$inf_k$ 和 sup_k 分别表示分组长度特征分量 l_k 的阈值下限和阈值上限，当 inf_k 与 sup_k 相等时，l_k 取固定值，否则取范围值。

特别地，L 可取位置分组长度变量，所谓的位置分组长度变量是指某个分组位置处的数据分组长度，该数据分组长度事先未知，而只能进行动态提取和确定。

定义 5　分组序列连续性

在流中指定的流方向 D_F 上的数据分组序列在连续区间位置 X 上不间断的一一出现并且满足相应的分组序列方向特征 D_P 和分组序列分组长度特征 L，则称为分组序列满足连续关系。分组序列连续关系记为 $R_c(D_F, X, D_P, L)=\{r_{cc}, r_{cv}\}$，其中 r_{cc} 表示分组序列连续，r_{cv} 表示分组序列不必连续。

定义 6　分组序列有序性

在流中指定的流方向 D_F 上的数据分组序列在指定的分组序列位置（离散序列位置或连续区间位置）X 上按照指定的先后顺序依次出现并且满足相应的分组序列方向特征 D_P 和分组序列分组长度特征 L，则称为分组序列满足有序关系。分组序列有序关系记为 $R_s(D_F, X, D_P, L)=\{r_{ss}, r_{sv}\}$，其中 r_{ss} 表示分组序列有序，r_{sv} 表示分组序列不必有序。

定义 7　分组长度分布特征

分组长度分布特征定义为带有效负载的抽样数据分组序列在流中指定的流方向 D_F、指定的分组序列位置 X 上应存在一个长度为 N 的数据分组子序列并满足分组序列方向特征 D_P、分组序列分组长度特征 L、分组序列连续性 R_c 和分组序列有序性 R_s 约束，记为 $F(D_F, X, D_P, L, R_c, R_s, N) = D_F X D_P L R_c R_s$。

定义 8　分组长度统计签名

分组长度统计签名定义为应用协议类型 C 已知的加密流量的分组长度分布特征 F、统计量 T 以及统计量 T 的期望值 T_e，记为 $P(F,T,T_e,C)$。其中，$T_e=[t_{inf}, t_{sup}]$。典型的分组长度统计签名如表 1 所示，对于不同的统计签名，T_e 表示数据分组子序列的长度或单个分组长度。

表 1　　**典型的分组长度统计签名**

统计签名 P	统计量 T	含义
分组长度序列签名（LenSeqSig）	符合 F 的数据分组子序列的分组计数	L 指定为分组长度序列；在流中指定的 D_F、指定的 X 上存在一个分组长度与 L 一一对应的数据分组子序列，并满足指定的 D_P、R_c、R_s 约束
分组长度集合签名（LenSetSig）	符合 F 的数据分组子序列的分组计数	L 指定为分组长度集合；在流中指定的 D_F、指定的 X 上存在一个数据分组子序列，其分组长度皆包含于 L，并满足指定的 D_P、R_c、R_s 约束
等分组长度序列签名（SLenSeqSig）	符合 F 的数据分组子序列的分组计数	L 指定为单个位置分组长度变量；在流中指定的 D_F、指定的 X 上存在一个数据分组子序列，其分组长度皆等于 L，并满足指定的 D_P、R_c、R_s 约束
同向分组序列签名（SDirSeqSig）	符合 F 的数据分组子序列的分组计数	D_P 指定为单个分组方向；在流中指定的 D_F、指定的 X 上存在一个数据分组子序列，其分组方向皆为 D_P，并满足指定的 R_c、R_s 约束，此处 L 无定义
分组长度均值签名（LenAvgSig）	分组长度均值	L 指定为单个分组长度；在流中指定的 D_F、指定的 X 上的数据分组序列的分组长度均值与 L 相匹配，并满足指定的 D_P 约束，此处 R_c、R_s 无定义
同向连续分组长度和值签名（DirLenSumSig）	同向连续分组长度和值	L 指定为单个分组长度；在流中指定的 D_F、指定的 X 上的第 N 轮同向连续数据分组的分组长度和值与 L 相匹配，并满足指定的 D_P 约束，此处 R_c、R_s 无定义。其中，流方向的改变标志着某轮同向连续数据分组的结束

定义 9 分组长度统计签名变量分组位置

分组长度统计签名可以引用其他的分组长度统计签名定义其分组序列位置。相对于当前分组长度统计签名所引用的分组长度统计签名的命中位置、在某个流方向上的带有有效负载的抽样数据分组序列的偏移位置编号，称为分组长度统计签名变量分组位置，记为 $X'=\{x'_i,\cdots,x'_j|1\leqslant x'_k\leqslant N,1\leqslant i\leqslant k\leqslant j\leqslant N\}$。其中，$x'_k$ 表示单个数据分组的偏移位置编号；N 表示所在流方向上可取的最大偏移位置编号。

分组长度统计签名决策模型

零假设 H_0：加密流量应用协议类型为 C。

备择假设 H_1：加密流量应用协议类型不为 C。

检验值 z：分组长度统计签名 P 的统计量 T。

显著性水平：α

临界值：$z_{1-\alpha/2}=[\inf(T_e)(1-\frac{\alpha}{2})+0.5]$，

$z_{1+\alpha/2}=[\sup(T_e)(1+\frac{\alpha}{2})+0.5]$，[] 表示取整。

拒绝域 Z_{rej}：$[0,z_{1-\alpha/2})\cup(z_{1+\alpha/2},+\infty)$

接受域 Z_{acc}：$[z_{1-\alpha/2},z_{1+\alpha/2}]$

观测值：z_{obs}

决策规则 $H(P)$：

$$\begin{cases}\text{True}(\text{接受零假设}H_0), & z_{obs}\in Z_{acc}\\ \text{False}(\text{拒绝零假设}H_0), & z_{obs}\in Z_{rej}\end{cases}$$

定义 10 DFI 特征

满足一定的逻辑关系 R_l 的多个分组长度统计签名 P 的集合，称为 DFI 特征，记为 $FF(\{P_i\};R_l)$。其中，逻辑关系 R_l 支持 AND、OR 和逻辑表达式，缺省为 AND；逻辑表达式由 AND、OR 和分组长度统计签名 P 的编号组成。

定义 11 半流关联特征

已识别应用协议类型 C 的流的源（或目的）IP、源（或目的）端口 *port* 和指定的传输协议 *tp* 组成的二元组或三元组称为半流关联特征，记为 *RF*(IP, *port*, *tp*, *C*)。其中，由 IP、端口和传输协议组成的三元组称为强关联特征，而 IP 和传输协议组成的二元组称为弱关联特征；半流关联特征 *RF* 中缓存有关联的应用协议类型 C。

已识别应用协议类型的流的半流关联特征 *RF* 通过散列运算生成关联半流表（RT, relational table），后续可通过提取的强关联特征直接进行关联查表确定流的应用协议类型，或通过提取的弱关联特征进行预过滤以筛选出需根据指定了该弱关联特征的规则进行后续识别的流。

定义 12 单数据分组特征

单数据分组特征定义为在流中指定的流方向 D_F、指定的分组位置 X 和指定的分组方向 D_P 上的单个数据分组应满足的关键字特征、分组长度特征、端口特征、IP 地址特征或半流关联特征等特征签名 *sig*，并且多个特征签名之间满足一定的逻辑关系 R_l，记为 $PF((D_F, X, D_P, \{sig\}); R_l)$。其中，逻辑关系 R_l 支持 AND、OR 和逻辑表达式，缺省为 AND；逻辑表达式由 AND、OR 和特征签名 *sig* 的编号组成。

定义 13 提升型 DPI 规则

由规则头 *HD*、单数据分组特征 *PF* 和可选的关联动作 *ACT* 组成的应用识别规则，称为提升型 bDPI（boosting DPI）规则，并记为 *DR*(*HD*, *PF*, *ACT*)。其中，规则头 *HD* 包括规则编号、应用协议类型 C、传输协议 *tp*、优先级 *prio* 等；关联动作 *ACT* 指定在规则命中时应提取并添加到关联半流表 *RT* 中的半流关联特征 *RF*。

定义 14 DFI 规则

由规则头 *HD*、单数据分组特征 *PF* 和 DFI 特征 *FF* 组成的应用识别规则，称为 DFI 规则，并记为 *SR*(*HD*, *PF*, *FF*)。其中，规则头 *HD* 包括规则编号、应用协议类型 C、传输协议 *tp*、优先级 *prio* 等；单数据分组特征 *PF* 为在验证 DFI 特征 *FF* 之前应首先满足的预过滤条件。

DFI 决策模型

预过滤条件 $H(PF)$:

$$\begin{cases}\text{True}, & (DPI(sig_1),\cdots,DPI(sig_n))\in R_l\\ \text{False}, & (DPI(sig_1),\cdots,DPI(sig_n))\notin R_l\end{cases}$$

流统计特征条件 $H(FF)$:

$$\begin{cases}\text{True}, & (H(P_1),\cdots,H(P_n))\in R_l\\ \text{False}, & (H(P_1),\cdots,H(P_n))\notin R_l\end{cases}$$

决策规则 $H(DFI)=H(PF)\wedge H(FF)$

4 加密流量应用识别算法

4.1 PLSSI 匹配算法

分组长度统计签名匹配算法 PLSSI(packet length statistical signature identification)基于分组长度统计签名决策模型实现，其伪代码如下文所示。

算法输入：分组长度 l，分组方向 d，分组长度统计签名 P、分组长度统计签名 P 的匹配状态 S_P 和流在各方向上的当前分组位置 *cp*[]。其中，匹配状态

S_P 包括当前分组位置 x、统计量 T 的当前分组长度统计量 l'、统计量 T 的当前分组计数 n、位置分组长度变量的当前值 l''、连续性状态 r_c、有序性状态 r_s、当前识别状态 Q_P（PENDING、HIT、FAILED）。

算法输出：带更新状态的分组长度统计签名 P。

算法描述：

1) **If** d is in the direction of D_F **Then**
2) **If** $\boldsymbol{X} \neq$ variable position **Then**
3) $x \leftarrow cp[D_F]$
4) **Elif** Q_P of referred P' = HIT **Then**
5) $x \leftarrow x + 1$
6) **End If**
7) **If** $\boldsymbol{L}$ is variable length and $\boldsymbol{L} = x$ **Then**
8) $l'' \leftarrow l$
9) **End If**
10) **If** $x \in \boldsymbol{X}$ and $d = D_P[x]$ **Then**
11) **If** P is of bytes statistic type **Then**
12) Update l' with l
13) $Z_{obs} \leftarrow l'$
14) **Elif** $l \in \boldsymbol{L}[x]$ **Then**
15) Update r_c 、r_s and n
16) $Z_{obs} \leftarrow n$
17) **End If**
18) **End If**
19) **If** $x = \max(\boldsymbol{X})$ **Then**
20) **If** $Z_{obs} \in \boldsymbol{Z}_{acc}$ **Then**
21) $Q_P \leftarrow$ HIT
22) **Else**
23) $Q_P \leftarrow$ FAILED
24) **End If**
25) **End If**
26)**End If**

4.2　HMETI 识别算法

HMETI（hybrid method for encrypted traffic identification）加密流量应用识别算法基于 DFI 决策模型实现，分为预处理和识别 2 个阶段。其中，预处理阶段根据 bDPI 规则和 DFI 规则的单数据分组特征生成包括多模式匹配状态机和散列表的 DPI 引擎，而识别阶段则首先利用 DPI 引擎筛选出命中了预过滤条件的 DFI 规则集，然后基于 HTSDM 模型对初步命中的 DFI 规则进行 DFI 特征的验证。通常情况下，需要对目标流进行多次识别，并且最多只处理流的前 N（通常取 N=60）个带有效负载的数据分组。HMETI 算法的伪代码如下所示。

算法输入：规则集合 SET，分组上下文 *pkt*，流节点 *fn*，关联半流表 *RT*。其中，规则集合 SET 包括 bDPI 规则 *DR* 和 DFI 规则 *SR*；分组上下文 *pkt* 包括当前数据分组的分组长度 *l*、分组方向 *d*、载荷 *payload* 和五元组 *tuple* 等；流节点 *fn* 为会话流表节点，包括流统计子节点链表 *fsnlist*、流在各方向上的当前分组位置 *cp*[]、流识别状态 Q_F、流应用协议类型 *cid* 等；流统计子节点 *fsn* 与 DFI 规则 *SR* 相对应，包括 DFI 规则 *SR* 的各个分组长度统计签名 P 的匹配状态 S_P 和规则识别状态 Q_R。

算法输出：带有更新状态的流节点 *fn*。

算法描述：

Ⅰ）Preprocessing：

1) **For** *rule* in *SET* **Do**
2) **For** *sig* in *rule.PF* **Do**
3) **If** typeof(*sig*) = KEYWORD **Then**
4) **If** *sig* ≠ short signature **Then**
5) SigSet[*rule.hd.tp*][KEYWORD] ← *sig*
6) **End If**
7) **Else**
8) SigSet[*rule.hd.tp*][typeof(*sig*)] ← *sig*
9) **End If**
10) **End For**
11) **If** typeof(*rule*) = *SR* **Then**
12) **For** *pat* in *rule.FF* **Do**
13) Compute $\boldsymbol{Z}_{acc}$ acc. to *HTSDM* Model
14) **End For**
15) **End If**
16) **End For**
17) L4ProtoSet ← {TCP, UDP}
18)SigTypeSet←{KEYWORD, PKTLEN, PORT, IP,RF}
19) **For** *tp* in L4ProtoSet **Do**
20) **For** *sigtype* in SigTypeSet **Do**
21) **If** *sigtype* = KEYWORD **Then**
22) MultiPatMatch[*tp*]←Eng_Build (SigSet [*tp*][*sigtype*])
23) **Else**
24) HashTable[*tp*][*sigtype*]←Hash(SigSet [*tp*][*sigtype*])
25) **End If**
26) **End For**

27) **End For**

28) Dpi_Engine ⇐ MultiPatMatch[], HashTable[][],*RT*

Ⅱ）Identifying:

29) Update *fn.cp*[] with *pkt.d*

30) PreHitDistinctRules ← Dpi_Engine(*pkt*)

31) **For** *rule* in PreHitDistinctRules **Do**

32) **If** *rule.PF* has short *sig* **Then**

33) *rule.PF*.sigs.Hit ← Fixed_Position_Match(*sig*)

34) **End If**

35) **If** Eval_PF(sigs.Hit,R_l,*pkt*,*fn*) = HIT **Then**

36) FilteredRules[typeof(*rule*)] ← *rule*

37) **End If**

38) **End For**

39) **If** *rule* = Highest_Priority(FilteredRules[*DR*]) **Then**

40) **If** exists(*rule.ACT*) **Then** *RT* ⇐ Hash(*rule.ACT.RF*,*pkt*)

41) *fn*.Q_F ← HIT; *fn.cid* ← *rule.hd.C*

42) **Return**

43) **End If**

44) **For** *rule* in FilteredRules[*SR*] **Do**

45) **If** *rule* ∉ *fn.fsnlist.Rules* **Then**

46) *fn.fsnlist* ⇐ Create_Node(*rule*)

47) **End If**

48) **End For**

49) **For** *node* in *fn.fsnlist* **Do**

50) **If** *node.rule*.Q_R = PENDING **Then**

51) **For** *pat* in *node.rule.FF* **Do**

52) PLSSI(*l*,*d*,*pat*,*pat*.S_P,*fn.cp*)

53) **End For**

54) *node.rule*.Q_R ← Eval_FF(*pats*.S_P,R_l)

55) **If** *node.rule*.Q_R = HIT **Then**

56) *fn*.Q_F ←HIT; *fn.cid* ← *node.rule.hd.C*

57) **Break**

58) **End If**

59) **End If**

60) **End For**

5 实验与结果分析

为了对前文所述的加密流量应用识别方法的有效性进行评估，本文在 Linux 平台上实现了 HMETI 应用识别引擎库，并基于 Libpcap 和 readline 库实现了相应的驱动测试平台 TrafficBench，支持规则集配置、报文回放、识别结果统计报表、基于识别结果的报文过滤及导出等功能。

5.1 评价指标

网络流量应用识别方法准确性的评价指标主要有误报率、精确率、召回率、总体准确率和总体误报率等几种。此处讨论的网络流量应用识别方法包括应用协议识别算法和对应的规则集合。

误报（FP, false positive）是指将本不属于某类应用的流量识别为该类应用；漏报（FN, false negative）是指将本属于某类应用的流量识别为其他类型应用；真报（TP, true positive）是指将属于某类应用的流量识别为该类应用。

表 2 识别方法评价指标的符号约定

符号	说明
N	应用协议种类的数量
i	第 i 类应用协议
FP_i	误报为第 i 类应用协议的流量
FN_i	漏报第 i 类应用协议的流量
TP_i	真报为第 i 类应用协议的流量

假定测试样本集由 N 类应用的流量构成，使用网络流量应用识别方法对该测试集进行识别，按照表 2 给出的符号约定，第 i 类应用协议识别的准确性评价指标定义如下。

误报率（FPR, false positive rate）

$$FPR_i = \frac{FP_i}{FP_i + TP_i + FN_i} \tag{1}$$

精确率（precision）

$$PR_i = \frac{TP_i}{TP_i + FP_i} \tag{2}$$

召回率（recall）

$$REC_i = \frac{TP_i}{TP_i + FN_i} \tag{3}$$

全部应用协议识别的总体准确性评价指标定义如下。

总体准确率（overall accuracy）

$$ACC = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} (TP_i + FN_i)} \tag{4}$$

总体误报率（overall FPR）

$$FPR=\frac{\sum_{i=1}^{N}FP_i}{\sum_{i=1}^{N}\left(TP_i+FN_i\right)} \tag{5}$$

如果上述定义采用不同的统计粒度（例如流、分组个数、字节数等），则可得到网络流量应用识别方法在不同维度的应用协议识别准确性评价指标。

5.2　数据集

本文选取了支持协议加密/混淆的 P2P 应用 BitTorrent（简称 BT）和 eMule 评估前文所述加密流量应用识别方法的有效性，其中，BitTorrent 客户端选用 BitTorrent V7.6.1 和 uTorrent V3.3，eMule 客户端选用 eMule V0.50a 和 easyMule V1.2.0，并且开启了协议加密/混淆功能。

评估所用的数据集分别单独按照不同应用捕获自实验室环境，如表 3~表 5 所示。表 3 中的 BitTorrent 和 eMule 数据集 1 由 19 个 Trace 文件组成，每个 Trace 为 BitTorrent 或 eMule 产生的全部 TCP 和 UDP 混合流量，包括 Web 流量、明文数据流量和加密流量，并且滤除了 DNS、ARP 等无关流量。表 4 中的 BitTorrent 和 eMule 数据集 2 分为训练集和测试集 2 部分，其中，训练集为人工分类和标注的 TCP 加密数据流，而测试集为从数据集 1 中过滤出的无法通过 DPI 识别的全部 BitTorrent 或 eMule TCP 数据流，具体的 bDPI 规则如下文表 6 所示。表 5 中的背景流量数据集 3 共计 286 个 Trace 文件，分别对应各种常见的加密应用和普通应用（或协议），其中加密应用占据了绝大部分流量。

表 3　BitTorrent 和 eMule 数据集 1

应用协议		流数	分组数	字节数/byte
BitTorrent	TCP	7 773	335 233	264 750 028
	UDP	5 684	115 830	68 450 154
	Total	13 457	451 063	333 200 182
eMule	TCP	5 791	309 770	215 663 863
	UDP	6 055	52 157	5 754 177
	Total	11 846	361 927	221 418 040

表 4　BitTorrent 和 eMule TCP 数据集 2

应用协议		流数	分组数	字节数/byte
训练集	BitTorrent	36	9 527	8 681 142
	eMule	36	10 739	8 970 280
测试集	BitTorrent	101	213 624	187 753 631
	eMule	188	215 620	189 089 873

表 5　背景流量数据集 3

应用协议		流数	分组数	字节数/byte
FTP,HTTP(S),DNS, POP3(s),SSH, SIP,Skype,QQ, ppfilm,QVOD, Baofeng,Funshion, Vagaa,SoulSeek, Ares,WoW,迅雷	TCP	11 941	667 166	449 431 023
	UDP	13 875	398 106	134 878 660
	Total	25 816	1 065 272	584 309 683

为了模拟真实网络环境出口捕获流量的特性，例如本地主机 IP 分布和不同应用在本地主机上的分布，数据集的全部 Trace 统一进行了单个本地主机 IP 的重新映射处理。本地主机 IP 的映射方法如下：1）预设私有 IP 地址池 1 和 2，其中，IP 地址池 1 容量设置为 30，IP 地址池 2 容量设置为 200，并且 IP 地址池 1 为 IP 地址池 2 的子集；2）将每个 BitTorrent 或 eMule Trace 中的本地主机 IP 随机映射为 IP 地址池 1 中的某个私有 IP，将背景流量的每个 Trace 中的本地主机 IP 随机映射为 IP 地址池 2 中的某个私有 IP，并保证不同应用类型 Trace 之间的本地主机 IP、非知名端口和传输协议三元组无冲突。经过重映射处理之后，数据集的 Trace 包含多个本地主机，并且每个本地主机 IP 对应一种或多种应用，与实际网络流量分布模型相一致。

5.3　规则集

实验采用的规则集包括 BitTorrent、eMule 和 Web 的相应 bDPI 规则和 DFI 规则。

由于 BitTorrent、eMule 和 Web HTTP 都属于开源协议，其单数据分组特征较易于分析和提取，具体的 bDPI 规则如表 6 所示。为方便计算，bDPI 规则的单数据分组特征的关键字特征采用正则表达式语法描述，在实际解析和预处理时，应分离出正则表达式的所有因子字符串并保留其在数据分组内的位置信息和字符串之间的逻辑关系。作为典型的 P2P 应用，BitTorrent 和 eMule 使用 UDP 和单个端口与大量的节点进行 DHT/Kad 网络通信以执行查找资源、维护节点联通性等功能或进而进行基于 UDP 的数据传输，因此，通过将相应的 bDPI 规则关联动作设定为源强关联以直接识别该类 UDP 流量。由于 BitTorrent 在进行 TCP 加密数据传输时必然伴随着与 Tracker 进行通信，因此，通过将相应的 bDPI 规则关联动作设定为源弱关联可输出运行 BitTorrent 应用并可能进行加密数据传输的候选主机，该弱关联特征可作为进行 BitTorrent 加密流量识别的先决条件。

表 6 BitTorrent、eMule 和 Web 应用协议的 bDPI 规则

应用协议规则头	*PF*					*ACT*	来源
	特征类型	D_F	X	D_P	*sig*		
Web-HTTP-1	Keyword	d_b	2	d_d	/^http/(1\.0|1\.1)/	×	HTTP Response
BitTorrent-TCP-1	Keyword	d_b	1	d_u	/^.*Torrent protocol/	×	BitTorrent Handshake
BitTorrent-TCP-2	Keyword	d_b	1	d_u	/^GET .*\?info_hash=/	源弱关联	BitTorrent Tracker
BitTorrent-UDP-3	Keyword	d_b	1	d_u	/^ d1:ad2:id20/	源强关联	BitTorrent DHT
eMule-UDP-1	Keyword Pktlen	d_b	1	d_u	/^\xe4(\x20|\x21)/ [35,35]	源强关联	eMule Kad
eMule-TCP-2	Keyword Pktlen	d_b	1	d_u	/^\xe3/ (Data[1:4]+5)==LoadLen	×	eMule OP

为了选择和提取 BitTorrent 和 eMule 的 DFI 特征，基于数据集 2 中的训练集样本和典型的分组长度统计签名，考察 BitTorrent 和 eMule 在双向流方向、上行流方向上的分组长度分布以及 BitTorrent 的同向连续分组长度和，统计结果如图 1~图 4 所示。

图 1 为分别从 BitTorrent 和 eMule 的 36 条加密数据流中抽取的双向流方向上的前 30 个数据分组的分组长度分布散点图，其中，X 轴表示双向流方向上的分组位置，Y 轴表示数据分组长度，正值表示分组方向为上行，而负值则表示分组方向为下行（坐标轴正负值含义下同）。由图 1 可知，BitTorrent 加密数据流的首分组分组长度介于 70~300，eMule 首分组分组长度介于 12~270，第 2 个分组的分组方向总为下行且分组长度介于 6~261，第 3 个分组的分组方向总为上行且分组长度介于 95~200，第 4 个分组的分组方向总为下行且分组长度介于 86~358。

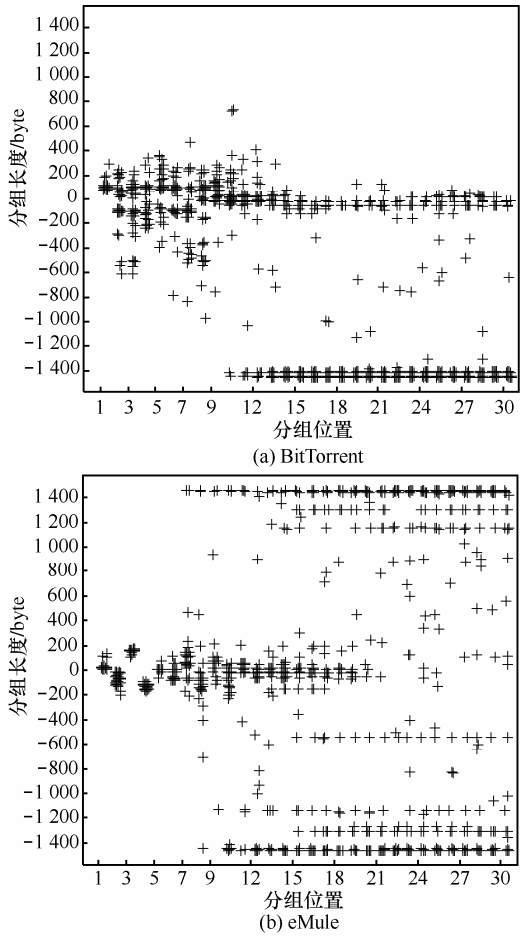

图 1 BitTorrent 和 eMule 加密流双向前 30 个数据分组长度分布

图 2 为 BitTorrent 的 36 条加密数据流上行流方向上的第 4~20 分组位置上分组长度小于 200 的数据分组的分组长度分布散点图，其中，X 轴表示流编号，Y 轴表示在对数坐标下的数据分组长度。由图 2 可知，分组长度 17 和 34 为频繁项并构成所有流的集合覆盖，这表明在 BitTorrent 加密数据流上行流方向的第 4~20 分组位置上至少存在 1 个分组长度等于 17 或 34 的数据分组。

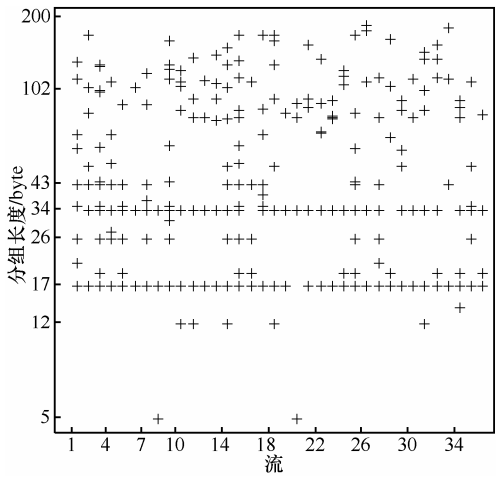

图 2 BitTorrent 加密流上行数据分组分组长度分布

图 3 为 eMule 的 36 条加密数据流上行流方向上的第 3~10 分组位置上分组长度小于 300 的数据分组的分组长度分布散点图，其中，*X* 轴表示流编号，*Y* 轴表示在对数坐标下的数据分组分组长度。由图 3 可知，分组长度 6、11 和 22 为频繁项并构成所有流的集合覆盖，这表明在 eMule 加密数据流的上行流方向的第 3~10 分组位置上至少存在 1 个分组长度等于 6、11 或 22 的数据分组。

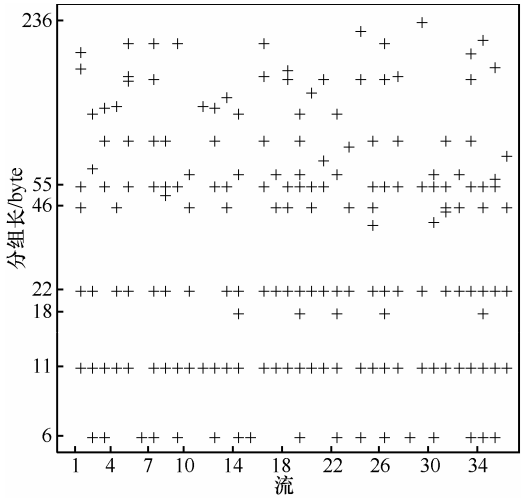

图 3　eMule 加密流上行数据分组分组长度分布

图 4 为从 BitTorrent 的 36 条加密数据流中抽取的上行和下行流方向上同向连续数据分组分组长度和的散点图，其中，*X* 轴表示上/下行连续交替位置，*Y* 轴表示同向连续分组长度和。由图 4 可知，在 BitTorrent 加密数据流的上行流方向的第 1~2 轮同向连续数据分组的分组长度和分别介于 95~610 和 5~640，下行流方向的第 1 轮同向连续数据分组的分组长度和介于 80~610。

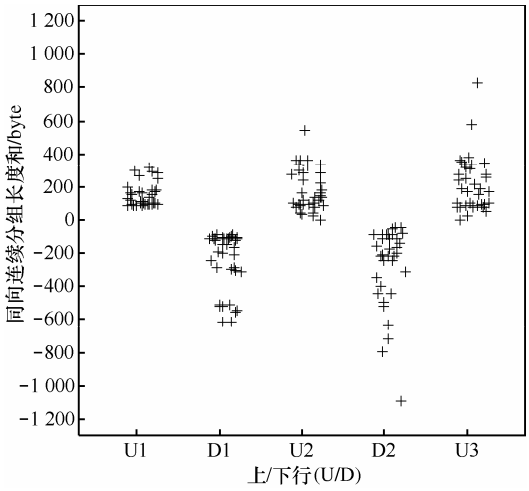

图 4　BitTorrent 同向连续数据分组的分组长度和

基于上述分析，可以得到 BitTorrent 和 eMule 的 TCP 加密协议 DFI 规则，如表 7 和表 8 所示。

5.4　实验结果

以数据集 3 作为背景流量，基于上述 BitTorrent 和 eMule TCP 加密协议 DFI 规则，取显著性水平 α=0.01，运用 HMETI 算法对数据集 2 的测试集样本进行加密流量识别，得到的识别结果如表 9 所示。由表 9 可知，BitTorrent 加密流量识别的字节精确率和召回率可达 98%以上，而 eMule 加密流量识别的字节精确率和召回率则分别为 100%和 99.9%；eMule 误报率为 0%，而 BitTorrent 误报率则相对较高，其字节误报率接近 2%。

表 7　BitTorrent TCP 加密协议 DFI 规则

特征	统计签名/类型	D_F	X	D_P	L/sig	R_c	R_s	N
PF	PktLen	d_b	1	d_u	[70,300]	×	×	×
FF	DirLenSumSig	d_b	[1,12]	d_u	[95,610]	r_{cc}	r_{ss}	1
	DirLenSumSig	d_b	[1,12]	d_u	[5,640]	r_{cc}	r_{ss}	2
	DirLenSumSig	d_b	[1,12]	d_d	[80,610]	r_{cc}	r_{ss}	1
	LenSetSig	d_u	[4,20]	d_u	17,34	r_{cv}	r_{sv}	1

表 8　eMule TCP 加密协议 DFI 规则

特征	统计签名/类型	D_F	X	D_P	L/sig	R_c	R_s	N
PF	PktLen	d_b	1	d_u	[12,270]	×	×	×
FF	LenSeqSig	d_b	3	d_u	[95,200]	r_{cc}	r_{ss}	1
	LenSeqSig	d_b	2,4	d_d	[6,261],[86,358]	r_{cv}	r_{ss}	2
	LenSetSig	d_u	[3,10]	d_u	6,11,22	r_{cv}	r_{sv}	1

为了降低 BitTorrent 加密流量误报率，考虑结合 bDPI 方法进一步加强预过滤条件进行优化，只对由 bDPI 判定为具有 BitTorrent 行为的主机进行后续加密流量识别，为此，在表 7 中的 DFI 规则的单数据分组特征 PF 中引入源弱关联特征并且联合表 6 中的 bDPI 规则 BT-TCP-2，利用数据集 1 中的 BitTorrent Trace 重复上述 BitTorrent 加密流量识别过程，得到的 BitTorrent 加密流量优化识别结果如表 10 所示。与表 9 所示的优化之前的识别结果相比，字节误报率显著降低，仅有 0.364%，字节精确率提高到 99.6%以上，而召回率保持不变。

表 9 BitTorrent 和 eMule 加密流量识别结果

指标		流	分组	字节
误报率	BT	10.619 5%	2.374 6%	1.867 5%
	eMule	0%	0%	0%
精确率	BT	84.415 6%	97.565 4%	98.098 7%
	eMule	100%	100%	100%
召回率	BT	64.356 4%	97.474 1%	98.188 3%
	eMule	96.276 6%	99.785 3%	99.904 0%

表 10 BitTorrent 加密流量优化识别结果

指标	流	分组	字节
误报率	2.884 6%	0.412 6%	0.364 0%
精确率	95.588 2%	99.576 8%	99.629 3%
召回率	64.356 4%	97.474 1%	98.188 3%

本文将 HMETI 算法与其他典型的加密流量应用识别方法进行了对比，结果如图 5 所示。其中，SPID、SLFC 和 K-K 算法分别由文献[3~5]提出。由图可知，无论是对于加密应用 BiTorrent 还是 eMule，本文提出的 HMETI 算法都具有比其他加密流量应用识别方法更高的识别准确率，这是由于 HMETI 算法引入了确定性抽样数据分组序列的位置、方向、分组长度和连续性、有序性等流统计特征，从而使该方法能够成功捕获加密应用在流坐标空间中独特的统计流量行为。

最后，考察 BitTorrent 和 eMule 产生的全部应用流量识别的总体准确性。利用 HMETI 算法和包括所有 bDPI 规则和 DFI 规则在内的规则集，取显著性水平α= 0.01，以数据集 3 作为背景流量，按照 4 种方法分别对数据集 1 进行完全流量识别，得到的总体准确率和总体误报率如图 6 所示。其中，X 轴表示识别方法，方法 1 使用传统 DPI 规则（无关联动作），方法 2 使用 bDPI 规则（带关联动作），方法 3 使用 bDPI 规则和未优化的 DFI 规则（不含源弱关联预过滤特征），方法 4 使用 bDPI 规则和优化的 DFI 规则（含源弱关联预过滤特征）；Y 轴表示在对数坐标下的总体准确率和总体误报率。

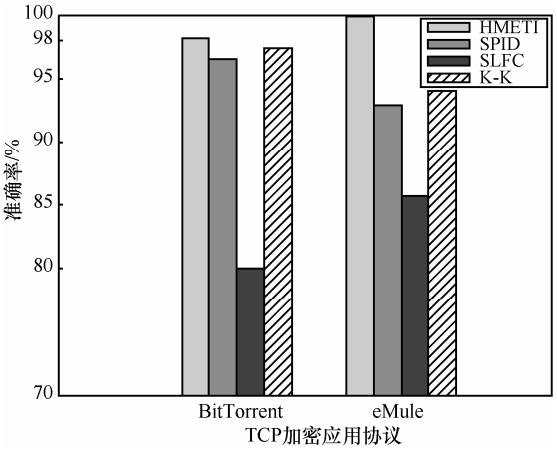

图 5 各算法的加密应用识别准确率对比

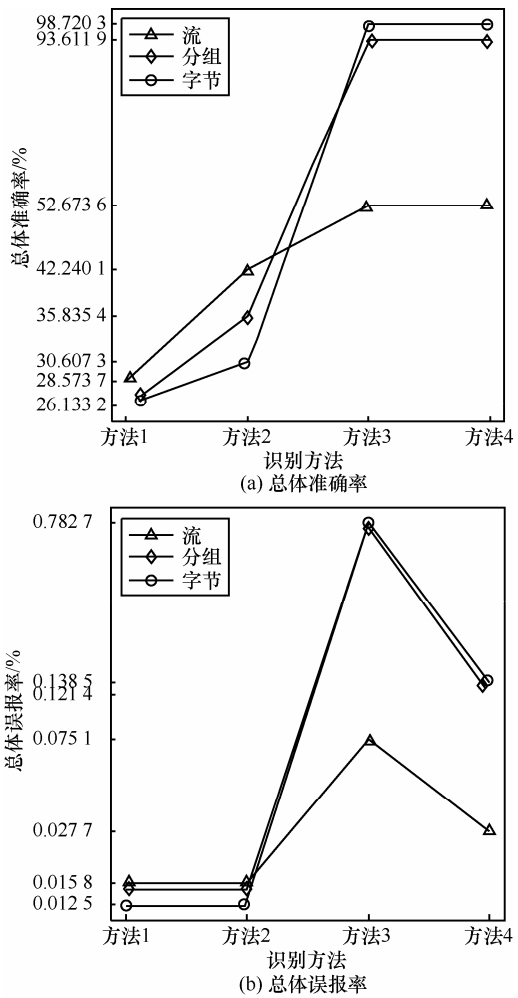

图 6 全部应用协议识别的总体准确率和总体误报率

由图 6 可知，传统 DPI 方法的字节总体准确率仅有 26.133 2%，这表明协议加密/混淆使传统 DPI 方法已经部分失效，而引入半流关联方法和 DFI 方法之后的字节总体准确率则逐步升高，在方法 4 时达到峰值，其字节总体准确率为 98.720 3%，这表明占据大部分比例且无 DPI 特征的 TCP 加密流量和 UDP 数据流量已被准确识别。另外，不同统计粒度（字节、分组与流）的总体准确率差别较大，这主要是由于在 P2P 类应用产生的大量会话中真正进行业务数据传输的流数量非常少，大部分为短会话或无效流，并且有部分加密数据流无法被识别。

同时，如图 6 所示，BitTorrent 和 eMule 流量识别的总体误报率非常低，对于识别方法 4，其在达到最高字节总体准确率的同时，字节总体误报率仅为 0.138 5%，具有最优的识别性能。从方法 2 到方法 4 时总体误报率抖动较大，原因是方法 3 引入了未优化的 DFI 方法导致了较高的加密流量识别误报，而方法 4 则使用了优化的 DFI 方法使加密流量识别的误报数量迅速下降。

6　结束语

本文基于加密应用在流坐标空间中的分组序列统计特征和典型的分组长度统计签名，提出了一种基于假设检验的加密流量应用识别统计决策模型 HTSDM，并给出了相应的基于 DPI 和 DFI 混合方法的加密流量应用识别算法 HMETI。最后，通过加密应用 BitTorrent 和 eMule 数据集评估了 HMETI 算法的有效性。实验结果表明，本文提出的加密流量应用识别方法可以达到接近 99%的字节总体准确率，并且仅有约 0.1%的字节总体误报率。

HMETI 算法依赖于数据分组在流中的位置和到达顺序等，因此需要对待识别流进行数据分组的去重传、分片重组、乱序重排等预处理，并且通常只应用于可靠有序的 TCP 加密流。同时，HMETI 算法对非对称路由[9]具有顽健性，对于无法获取完整流的应用场景，可使用单向流的分组序列统计特征。另外，由于采用了预过滤方法并且只需抽样识别流的少量数据分组，因此 HMETI 算法具有较低的计算复杂度并可应用到实时环境。

选取恰当的加密流量分组序列统计特征和分组长度统计签名是保证 HMETI 算法应用识别准确性的关键。目前，加密应用的流量统计特征和分组长度统计签名的提取主要是通过人工对捕获的大量流量 Trace 的分类、标注和分析进行，提取效率、特征的显著性和完整性都比较低。因此，下一步的研究工作将是利用数据挖掘算法进行加密流量统计特征和分组长度统计签名的自动提取和验证。

参考文献：

[1] GOMES J V, INÁCIO P R M, PEREIRA M, *et al*. Detection and classification of peer-to-peer traffic: a survey[J]. ACM Computing Surveys, 2013, 45(3):1-40.

[2] MOORE A, ZUEV D, CROGAN M. Discriminators for use in flow-based classification[R]. Technical Report RR-05-13, ISSN 1470-5559, University of London, 2005.

[3] HJELMVIK E, JOHN W. Breaking and improving protocol obfuscation[R]. Technical Report No.2010-05, ISSN 1652-926X, Chalmers University of Technology, 2010.

[4] LU C N, HUANG C Y, LIN Y D, *et al*. Session level flow classification by packet size distribution and session grouping[J]. Computer Networks, 2012, 56(1):260-272.

[5] BAR-YANAI R, LANGBERG M, PELEG D, RODITTY L. Realtime classification for encrypted traffic[A]. Proceedings of 9th International Symposium on Experimental Algorithms (SEA 2010)[C]. Naples, 2010.373-385.

[6] ALSHAMMARI R, ZINCIR-HEYWOOD A N. Machine learning based encrypted traffic classification: identifying SSH and skype[A]. Proceedings of the 2009 IEEE Symposium on Computation Intelligence in Security and Defense Applications (CISDA 2009)[C]. Ottawa, 2009.1-8.

[7] DUSI M, ESTE A, GRINGOLI F, SALGARELLI L. Using GMM and SVM-based techniques for the classification of SSH-encrypted traffic [A]. Proceedings of the 44th IEEE International Conference on Communication(ICC' 09)[C]. Dresden, 2009.1-6.

[8] NGUYEN T, ARMITAGE G. A survey of techniques for internet traffic classification using machine learning[J]. IEEE Communications Surveys & Tutorials, 2008, 10(4):56-76.

[9] CROTTI M, GRINGOLI F, SALGARELLI L. Impact of asymmetric routing on statistical traffic classification[A]. Proceedings of the 7th IEEE Global Communications Conference (GLOBECOMM 2009)[C]. Honolulu, 2009.1-8.

作者简介：

高长喜（1978-），男，山东嘉祥人，北京天融信公司博士后、系统架构师，主要研究方向为网络与信息安全、高性能安全网关、网络流量分类和应用协议识别。

吴亚飚（1971-），男，福建尤溪人，北京天融信公司总工程师、高级工程师，主要研究方向为安全网关技术架构、操作系统安全、内容/数据安全、安全硬件加速技术等。

王枞（1958-），女，北京人，北京邮电大学教授、博士生导师，主要研究方向为智能信息处理、网络信息安全、容灾备份等。