doi:10.3969/j.issn.1000-436x.2013.z2.009

# 基于决策树的 P2P 节点识别技术研究

董永苹[1]，余翔湛[2]，吴刚[1]
（1. 哈尔滨工业大学 网络与信息中心，黑龙江 哈尔滨 150001；2. 哈尔滨工业大学 计算机科学与技术学院，黑龙江 哈尔滨 150001）

**摘 要**：通过对 P2P 应用的长期研究，根据 P2P 节点自身的特点选取了其中典型的特征属性，并提出了一种基于决策树模型的 P2P 节点识别方法。由于该方法是统计分析传输层数据分组的特征，因此对于采用加密或非加密的 P2P 应用的网络节点识别均有效。通过实验验证，与基于端口和基于负载特征的流量监测方法相比，所提出的方法体现出了较高的准确率和较低的漏报率及误报率。

**关键词**：P2P；流量识别；决策树；流量特征
**中图分类号**：TP393.06 **文献标识码**：A **文章编号**：1000-436X(2013)Z2-0040-07

# Implementation of P2P nodes detection based on decision tree

DONG Yong-ping[1], YU Xiang-zhan[2], WU Gang[1]
(1. Computer Network Center, Harbin Institute of Technology, Harbin 150001, China;
2. School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

**Abstract:** A P2P nodes detection method based on decision model was proposed by a long time of observation. As this method is a statistical analysis of the transport layer packet characteristics, identification of the network node for encrypted or unencrypted P2P applications is effective. Experiment shows that this method has higher accuracy and lower false positive rate and false negative rate.

**Key words:** P2P; traffic identification; decision tree; traffic characteristic

## 1 引言

随着信息技术的高速发展，P2P(peer-to-peer)以其独特的技术优势被大量用户所青睐，P2P 技术已经被普遍应用于文件共享、语音服务、即时通信和流媒体传输等各个领域。然而由于 P2P 的广泛应用，P2P 流量占用了大量带宽，导致网络的承载负担日益加重。据统计，P2P 的流量自 2006 年起就已经占网络总流量的 60%以上，成为网络带宽的最大消耗者；另一方面，P2P 的恶意流量危害网络安全的形势也非常严峻。因此，如何有效地识别和控制 P2P 的流量正逐渐成为当今研究的焦点。

目前根据采用的识别方法不同，P2P 流量的识别又可分为端口识别技术、深层数据分组识别(DPI, deep packet inspection)技术和基于行为特征的流量识别技术。由于 P2P 技术常采用动态端口和端口跳跃技术来躲避检测，因此基于端口的识别方法已经无法满足流量监控的需求。文献[1]中指出，端口识别技术在对 P2P 流量进行分类过程中，有 30%~70%的流量未能正确识别。深层数据分组识别技术则主要是根据流量载荷与特征码进行匹配来识别具体的 P2P 应用，在大多数情况下，DPI 技术的识别准确性和可靠性均非常高[2]，但是针对负载加密的流量 DPI 技术则不再有效。文献[3]表明，利用 DPI 技术来识别负载加密流量(例如 eMule)的准确率非常低，甚至达不到 50%。基于行为特征识别技术的主要思想则是利用传输层数据流的统计信息来辨别 P2P 应用，它不依赖于应用层负载信息，而是一段时间内的流量统计特征来建立分类模型识别 P2P 流量。该识别技术在实际应用中存在的主要问题是，能够粗粒度识别 P2P 流量，但很难将 P2P 数据流细化到具体协议。

在 P2P 网络中，每个节点(peer)都承担着 2 种功能角色：客户端和服务器端的特性。与普通应用

收稿日期：2013-09-12

的客户节点不同，P2P 节点在下载资源的同时也在作为服务器为其他节点提供服务功能，这使得 P2P 应用的传输层与其他的网络应用(例如，DNS、FTP、Mail、Web 服务等)有着非常大的区别。因此，当前很多算法的研究正是基于 P2P 的双重角色特性。Fivos 等人便是利用 P2P 节点的双重特性和 P2P 网络的网络直径来进行区分 P2P 流量[4]。

文献[5]指出 P2P 网络中非常多的 P2P 应用既采用 TCP 协议，又同时采用 UDP 协议，但是其他非 P2P 应用则很少采用双重协议来进行信息交互。文献[6]认为 P2P 节点的双重角色特征必定导致 P2P 节点既具有发起连接又具有回应连接，因此用主机的反向 TCP 连接和正向 TCP 连接的比来判断是否为 P2P 节点。由于 P2P 的覆盖网络比较大，所以 P2P 节点对应的远端主机分布比较广，Bolla 等人则根据远端主机离散度来对 P2P 节点和非 P2P 节点进行区分[7]。

随着数据挖掘技术的引入，多种机器学习方法被应用到流量识别技术中。文献[8,9]利用机器学习的方法根据数据流的特征对加密流量进行应用识别。文献[10]采用了朴素贝叶斯分类算法进行网络流量的分类；文献[11]采用基于支持向量机(SVM)来识别 P2P 和非 P2P 流量。文献[12]则利用决策树、朴素贝叶斯、SVM 和反馈神经网络 4 种机器学习方法对相同数据进行分类，总结出决策树在分类实验中表现最优。

## 2　P2P 节点特征分析

在基于行为特征的流量识别技术中，主要是通过统计分析网络中数据流的特性来对流量进行分类识别，本文便是利用该技术通过 P2P 节点所具有的特性来区分 P2P 节点和非 P2P 节点的。

### 2.1　相关行为的定义

首先为了描述的需要，本文给出一些概念的定义及形式化描述。

基于数据流的特征，作者按照五元组<srcIP, destIP, srcPort, destPort, Protocol>来将不同的数据流区分，并将每个流分成双向 TCP 或 UDP 流，其中，srcIP 和 srcPort 分别为连接发起端的 IP 地址和端口，同理，destIP 和 destPort 为连接接收端的 IP 地址和端口，Protocol 则代表使用的协议（TCP 协议或 UDP 协议）。其他定义如下所述。

正向流(posFlow)：针对网络节点 *N*，如果数据流 flow 为节点 *N* 首先发起，则认为该 flow 为节点 *N* 的正向流。

反向流(negFlow)：针对网络节点 *N*，如果数据流 flow 是由其他网络节点 *M* 首先发起并与节点 *N* 建立连接，则认为该 flow 为节点 *N* 的反向流(和节点 *M* 的正向流)。

长连接流(longConnFlow)：一个数据流从第一个数据分组开始至最后一个数据分组为止，如果持续的时间超过一个阈值(本文中为 200 s)，则认为该流为一个长连接流。

连接(connection)：针对一个双向流本文认为是有效连接(effective connection)，否则为无效连接(ineffective connection)。

一个网络节点 *N* 上所存储的信息分别为<ownIP, posTcpFlows, posUdpFlows, negTcpFlows, negUdpFlows, longConnFlows, posConns, negConns, negUpload, negDownload, ownPorts, distPorts, distIPs>。其中，ownIP 为本节点网络地址，posTcpFlows 和 posUdpFlows 分别为本节点上正向 TCP 流个数和 UDP 流个数，negTcpFlows 和 negUdpFlow 分别为反向 TCP 流个数和 UDP 流个数，longConnFlows 为本节点上长连接个数，posConns 和 negConns 则分别代表正向总的连接个数和反向总的连接个数(包括有效连接和无效连接)，ownPorts 则代表本网络节点作为服务器端时自身所开放的所有端口，distIPs 和 distPorts 代表本节点作为服务器端时所有连接的远端节点的 IP 地址和端口号。

### 2.2　节点特征的分析

实际网络中由于网络节点的动态特性，常常发生节点从非 P2P 状态转变为 P2P 状态(或由 P2P 状态转变为非 P2P 状态)的现象。在状态转换的过程中，节点上并发连接数一般情况下会有非常大的波动。图 1 显示了使用迅雷进行下载到退出迅雷整个过程中节点上并发连接数的变化情况，在开始时刻开启了迅雷软件并在 57 s 时进行搜索和下载资源，750 s 时退出迅雷。从图 1 中可以看到，在开启迅雷这个 P2P 软件并开始下载资源后，节点并发连接数从 62 迅速增加到 150，这主要是由于 P2P 软件需要连接大量客户端以确定哪些客户端可以作为自己的服务节点。在连接上足够的客户端后，节点会断开那些不需要的连接，使得并发连接数回到 P2P 节点的正常水平（图 1 中约为 90 个）。

由于 P2P 节点上并发连接数目比较多，而且节点之间传输数据量比较大，所以 P2P 节点在传输层

存在很多典型的特征[13]。本文中选取的用来区分 P2P 节点与非 P2P 节点的特征属性如表 1 所示。

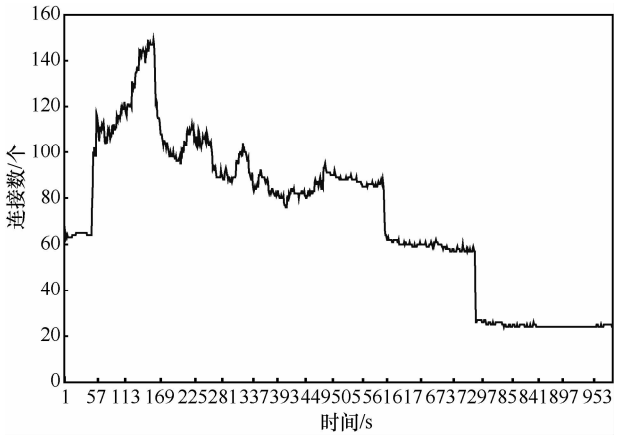


图 1 P2P 状态到非 P2P 状态转换过程中并发连接数的分布

**表 1 P2P 节点在传输层的特征属性**

| 节点特征 | 标号 | 公式描述 |
|---|---|---|
| 服务端口与远端 IP 数之比 | F1 | ownPorts/distIPs |
| 远端端口与远端 IP 数之比 | F2 | distPorts/distIPs |
| 长连接数所占比重 | F3 | longConn-Flows/(posTcpFlows+posUdpFlows+negTcpFlows+negUdpFlows) |
| 反向流中上传与下载量之比 | F4 | negUpload/negdownload |
| 正向流与反向流数之比 | F5 | (posTcpFlows+posUdpFlows)/(negTcpFlows+negUdpFlows) |
| 正向连接成功率 | F6 | (posTcp-Flows+posUdpFlows)/posConns |
| 服务率 | F7 | (negTcpFlows+negUdpFLows)/negConns |

1) 服务端口与远端 IP 数之比

对于普通 C/S 服务器，F1 值一般在(0, 1)范围内，这是由于服务器在本土通常仅开放一个或几个指定端口，所以 ownPorts 的值会比较小，而与该服务器节点同类的远端节点的 IP 数目却比较多，使得 distIPs 值比较大，所以 F1 的值通常非常小并且接近于 0。对于普通 C/S 客户端，F1 值常大于 1，主要原因是客户端与服务器的服务端口与远端 IP 数分布情况正好相反。在 P2P 应用中，对于主要采用 TCP 的 P2P 节点来说，F1 的值在 1 附近，主要原因是节点通常开放一个监听端口和一系列随机端口与网络中其他节点进行通信。而采用 UDP 流的 P2P 节点的 F1 的特征值就非常小且几乎为 0，因为采用 UDP 通信的 P2P 节点通常只开放一个端口，发起连接和接收连接都共用此端口。

2) 远端端口与远端 IP 数之比

在实际网络环境中，2 个 P2P 节点选择同一端口的概率非常小，因此 F2 的特性值近似为 1。在 C/S 应用中，F2 的值则通常大于 1，这主要是因为客户端通常会多次连接服务器，并且每次连接时客户端 IP 地址不变但端口却常发生变化；而对于 C/S 的客户端节点，远端端口和远端 IP 数与服务器端情况正好相反，导致其 F2 值通常在(0, 1)范围内。

3) 长连接所占比重

P2P 应用多体现为文件共享、流媒体和即时通信，因此数据流的连接时间通常比较长，与普通的 C/S 服务器相比，长连接所占比重更高，因此 P2P 节点上 F3 的值普通较大。

4) 反向流中上传与下载量之比

P2P 节点在网络中扮演着客户端和服务器端的双重角色，因此 P2P 节点通常都会上传和下载大量的数据文件，而 C/S 服务器的反向连接基本只用于上传数据而下载的数据则非常少，所以 P2P 节点 F4 特征值较 C/S 服务器要小很多。同时，C/S 客户端几乎不存在反向连接流，因此通过比较反向流的数目便能很容易使其与 P2P 节点区分开来。

5) 正向流与反向流个数之比

由于 P2P 节点具有双重角色，所以在通常情况下，正向流和反向流都比较多，其 F5 值通常在 1 附近浮动。针对普通 C/S 服务器，数据流的类型主要为反向流，因此 F5 的值通常非常小且近似为 0。针对 C/S 客户端，其几乎不存在反向连接流，区分流的方式同 4)。

6) 正向连接成功率

由于 P2P 节点具有明显的动态特性，所以其节点通常会存在大量的失败连接，然而普通 C/S 节点中正向连接的成功率却都比较高。因此，相对普通节点而言，P2P 节点的正向连接成功率 F6 的值会比较小。

7) 服务率

服务率主要是指一个节点向其他节点提供服务的情况，通过本节点对其他节点发起连接的回应率来表示。P2P 网络将负载分散到了各个节点，使得每个 P2P 节点提供服务的负担较小，所以 P2P 节点针对其他节点发起的请求通常都可以及时地做出回应，因此提供的服务率 F7 的值也比较高。而传统的 C/S 服务器的服务压力要远远大于 P2P 节点的服务压力，因此非常多的客户端都向同一服务器发起连接时，服务器很难回应所有客户端的连接请求，最终导致服务率 F7 值相对较低。

此外还存在一些其他相关的 P2P 特征属性，如 IP 地址的网段分布、非特权端口的使用、网络的规模、数据流中数据分组的平均分组大小和平均分组到达时间间隔等，在这里不再进行过多的分析。

## 3　决策树的分类模型

数据分类是数据挖掘的主要内容之一，主要是通过分析数据样本而产生关于类别的精确描述，同时分类技术解决问题的关键是构造分类器模型。目前分类方法有很多种，主要包括：决策树分类、贝叶斯分类、支持向量机分类、人工神经网络分类、K-近邻分类方法等。与其他分类方法相比，决策树分类方法有算法简单、计算量小、准确度高以及对训练样本依赖程度低等特点。在实际网络环境中 P2P 网络流分布受时间、地点等多种因素影响，通常具有一定的不稳定性，因此本文选用决策树分类模型作为 P2P 节点识别的研究方法来降低对未知样本空间预测的误报率和漏报率。

决策树(decision tree)是一种从无序和无规则的训练样本中推理出树形结构表示形式的分类方法。它有一个根节点、多个分枝节点以及叶子节点组成，其中，树的每个分支代表一个测试输出，每个叶子节点代表一个具体类别。C4.5 算法是机器学习中一个有影响、广泛使用的算法，是 QUINLAN J R 对 ID3 算法的改进。与 ID3 算法相比，C4.5 算法的准确度有了明显的提高。

### 3.1　C4.5 决策树模型

C4.5 算法的中心思想是利用信息熵原理在构建决策树时选择信息增益率最大的属性作为分裂属性[14]。ID3 算法以信息增益为标准，更加偏向于分类较多的属性，而 C4.5 算法利用信息增益率作为选择属性的依据，使得 ID3 算法中存在的这种问题得到了很好的解决。

#### 3.1.1　信息增益率

信息增益率是用来衡量给定的属性区分训练样本能力的一个重要指标。信息增益率最高的属性将被作为测试属性。决策树的生成过程就是使划分后的不确定性逐渐减小的过程。属性的信息增益率计算的步骤如下所示。

假设训练样本集合为 $S$，并且训练样本被分为 $k$ 类，即为 $C=\{c_1,\cdots,c_k\}$，此时集合 $S$ 的信息熵则为 $Entropy(S)=-\sum_{i=1}^{k}p(S_i)\log_b p(S_i)$，其中，$p(S_i)$ 是指 $S$ 中属于类别 $c_i$ 的比例，$b$ 是对数的底通常为 2、e 或 10。

假设属性集为 $A$，且 $A=\{A_1,A_2,\cdots,A_m\}$，选择 $A_j$ ($j\in[1,m]$)作为测试属性进行划分，并设 $Values(A_j)$ 为 $A_j$ 的值域，那么属性 $A_j$ 对样本集合 S 的信息增益为

$$Gain(S,A_j)=Entropy(S)-\sum_{v\in Values(A_j)}\frac{|S_v|}{|S|}Entropy(S_v)$$

其中，$|S|$ 为样本集合的元素个数，$|S_v|$ 为 $S$ 中属性 $A_j$ 的值为 $v$ 的子集元素个数。那么就可以计算出分裂信息，其中分裂信息的定义为

$$SplitInfo(S,A_j)=-\sum_{v\in Values(A_j)}\frac{|S_v|}{|S|}\log_b(\frac{|S_v|}{|S|})$$

此时便可求出属性 $A_j$ 的信息增益率。

$$Ratio(S,A_j)=\frac{Gain(S,A_j)}{SplitInfo(S,A_j)}$$

#### 3.1.2　决策树剪枝

为了避免决策树自身高度的无限制增长和数据的过度拟合，需要对决策树进行一定的剪枝处理，剪去决策树中可能降低预测准确率的分支。本文采用了后剪枝的方法，即在树生成后剪掉子树。决策树剪枝的具体过程如下。

对决策树中从根节点开始的所有非叶节点，计算其被剪后的误分类率。把训练样本集用作测试集，将置信区间的上限作为对误分类率的估计。对于一个即定显著性水平度 $\alpha$ (C4.5 算法中将 $\alpha$ 默认为 0.25)，错误总数服从二项分布，可得

$$\Pr\left[\frac{|f-q|}{\sqrt{q(1-q)/N}}>U_{1-\alpha}\right]=\alpha \qquad (1)$$

其中，$N$ 是样本实例的总数量，设 $E$ 为 $N$ 个样本实例中被错误分类的个数，那么 $f=E/N$ 为观察到的误差率，$q$ 为真实的误差率。令 $z=U_{1-\alpha}$ 为置信度 $\alpha$ 的标准差，通过式(1)可计算出 $q$ 的置信度上限，然后用此置信度上限为该节点误差率 $e$ 做一个悲观估计。

$$e=\frac{f+\frac{z^2}{2N}+Z\sqrt{\frac{f}{N}-\frac{f^2}{N}+\frac{z^2}{4N^2}}}{1+\frac{z^2}{N}} \qquad (2)$$

通过对剪枝前后 $e$ 值大小的判断，决定是否需要剪枝。整个修剪过程主要需考虑两方面因素：1）每个叶节点所含最少实例数，通过此数值可控制决策树的规模；2）显著性水平度($\alpha$)，通过该值可确定树的修剪程度。

### 3.1.3 C4.5 决策树的其他特征

C4.5 算法还对 ID3 算法中属性取值进行了完善，它既可以处理离散型的属性数据，也可以处理连续型的属性数据。在选择分裂属性时，对于离散型取值的属性，C4.5 的处理方法与 ID3 相同，按照该属性本身的取值个数进行计算；对于某一个连续型取值的属性 $A_j$，假设在某个节点上的数据集的样本数量为 $tot$，C4.5 将做以下处理。

算法将该节点上的所有数据集样本按照属性的具体取值由小到大排序，得到属性值的取值序列 $\{A_j[1], A_j[2],\cdots,A_j[tot]\}$。在取值序列中生成 $tot-1$ 个分割点，并且第 $i(0<i<tot)$ 个分割点的值则为 $V[i]=(A_j[i]+A_j[i+1])/2$,$V[i]$可以将该节点上的数据集划分为 2 个子集，从而完成对连续数据的离散化。通过分割点划分数据集的方式，C4.5 计算出每个分割点的信息增益率，并最终选择出信息增益率最大的分割点来划分数据集。

## 3.2 针对 P2P 节点的决策树模型构建

利用决策树进行分类主要分为两步：首先，利用训练集建立一棵决策树，构建出决策树模型；其次，利用生成的决策树模型对未知的数据样本进行分类，并进行准确率的预测。

在构建 P2P 节点的决策树模型中，网络节点的属性中信息增益率的大小决定着属性对分类的贡献性能，节点中信息增益率最高的属性被作为决策树的最高级分支节点。分支节点的每个取值对应一个分类子集，递归寻找和构建子集中的分支节点，直至完全生成一棵针对 P2P 节点分类的决策树。

针对 P2P 节点，由 C4.5 算法生成其决策树的具体过程如图 2 所示。

P2P 节点的决策树模型构建成功之后，需要利用一定的测试集对分类模型的准确率、漏报率和误报率进行预测评估。基于网络环境的自相似特性，当训练集和测试集数据量非常大并且具有非常高的代表性时，如果分类模型的准确率达到一定阈值并且误判率控制在一定范围内，该模型才具备应用到实际环境的基本条件。

输入：网络节点训练集 *T*，P2P 节点的属性集 *Attributes*
输出：一棵 P2P 节点的决策树
1) 创建决策树的 Root 节点
2) IF *T* 都属于同一类别 *C*，则返回 Root 为叶子节点，
并标记为 *C* 类;
3) IF *Attributes* 为空，则停止并返回;
4) FOR *Attributes* 中每一个属性 $A_j$ 根据 3.1.1 节的信息，
计算属性的信息增益率 $Ratio(T, A_j)$;
5) 选择具有最高信息增益率的属性 $A_i$，
将 $A_i$ 作为决策树的 Root 节点;
6) FOR 由 Root 生成的每个新的叶子节点
IF 该叶子节点对应的分类子集为 *T'*属于同一类别 *C'*
将该叶子节点标记为 *C'*类别;
ELSE
将叶子节点作为子树 Root 节点
并根据信息增益率递归分裂，goto 6);
7) 根据式(2)计算出每个节点的分类错误率，
并进行相应剪枝以纠正过度拟合问题
else if(*Attributes* 为空)
return 单节点树 Root，其预测类别为 Examples 中最普遍的类别;
else
选择 *Attributes* 中分类 Examples 能力最好的属性

图 2 由 C4.5 算法生成其决策树的具体过程

# 4 P2P 节点识别方法

## 4.1 实验数据集的获取

本文所采用的数据集均来自于哈尔滨工业大学的网络节点，所捕获的网络节点的应用类型主要是部分 P2P 应用节点(Thunder、PPLIVE、BitTorrent 和 eMule)和部分非 P2P 应用的服务节点(DNS server、Mail server、FTP server 和 Web server)。

为了避免多种应用的数据流所产生的影响，在一个网络节点上本文只运行一种上面提到的 P2P 应用或非 P2P 应用。在运行 P2P 应用或非 P2P 应用的各个网络节点上，笔者每天在固定的时间段(10:00~10:20，12:00~12:20，14:00~14:20, 19:00~19:20)抓取节点上的所有完整 IP 流，并分别保存。本文分时间段捕获网络节点上的数据流，其主要原因是考虑到一天中流量分布不均，因此分时间段对每个网络节点上的流量进行捕获；将抓捕分组的时间窗口设为 20 min，主要是因为若将窗口设置过小，那么统计时间太短可能会对实验结果造成较大误差，而时间窗口设置过大则不利于及时区分 P2P 节点与非 P2P 节点。

在多个被监控的网络节点上，按照上文描述的

时间段抓取数据分组，并且持续该抓捕分组过程 15 天。最后将捕获到的网络节点的数据流量按照该节点运行的应用进行分类存储，并作为训练和测试的样本集合，同时将每个节点上一个时间窗口的(即 20 min)的流量数据作为样本集合中的一个元素。

### 4.2　P2P 节点识别的过程

本文利用表 1 中选取的特征属性作为本次实验区分 P2P 节点与非 P2P 节点的依据。基于决策树的 P2P 节点识别方法的主要思想如下所示。

1) 首先将样本集分为训练集和测试集，每种应用的具体样本数量分布如表 2 所示。

2) 对样本元素中的数据流量根据数据流的五元组定义将数据报文分成双向 TCP 流或 UDP 流，然后对流的信息(表 1 中运用到的特征)进行统计。

3) 计算出表 1 中提到的节点特征属性所对应的特征值，并最终生成特征属性与节点信息的特征向量<F1, F2, …, F7, P2P or 非 P2P>。

4) 利用 3.2 节中的模型，针对训练集产生的特征向量构建决策树分类模型。

5) 利用已构建成功的决策树分类模型，对新的未知 P2P 节点和非 P2P 节点进行分类，并对准确率进行评估。

### 4.3　实验验证与结果分析

本实验中捕获到的各种应用类型及其对应的节点样本数目如表 2 所示。

按照表 2 中样本数的分布，本文利用构建的决策树模型进行分类研究，并对分类方法的准确率、漏报率和误报率进行了评估。为了比较本文的分类模型(C4.5-based)相对于基于端口(Port-based)识别技术和基于负载特征(Payload-based)识别技术的分类效果，作者也利用了基于端口的流量识别技术和基于负载特征的流量识别技术对样本集做了分类测试。上述几种 P2P 应用所对应的负载特征如表 3 所示，而非 P2P 应用本文则选用 L7-filter 中提供的匹配模式[15~17]。实验结果如图 3 所示。

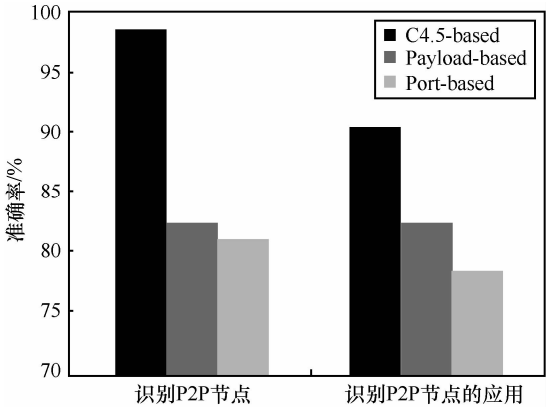


图 3　识别 P2P 节点与识别 P2P 节点的 P2P 应用的准确率状况

**表 3　　P2P 应用所对应的负载特征**

| 应用名 | 负载特征 |
|---|---|
| Thunder | '\x00\x00\x00\x16\x00\x00\x00\x6a\x01' |
| PPLIVE | '\x00\x00\x00\xe9\x03\x41\x41\x01\x98\xab\x01\x02' |
| BitTorrent | '\x13BitTorrent protocol' |
| eMule | '\xe3,\xc5,\xd4,\xe4,\xe5,\xf1' |

在图 3 中，与基于端口的流量识别技术和基于负载的流量识别技术相比，基于决策树模型的 P2P 节点识别的准确率有了显著的提升。针对网络节点类型的识别，该实验结果很好地说明了本文提出的分类模型的高效性和实用性。在识别 P2P 节点过程中，作者还针对 P2P 节点上具体 P2P 应用的识别准确率进行了测试与评估。图 3 结果显示，与识别 P2P 节点相比，本文中的分类模型在识别 P2P 节点的具体 P2P 应用时准确率有了明显的降低，出现该结果的主要原因是本文选择分类特征属性时主要考虑到的是 P2P 节点的共性特征，但对具体应用的特征并未做细化分析，因此对 P2P 应用的识别准确率造成了一定影响。

为了比较决策树算法相对于其他分类算法在本文中更具有适用性，作者还利用了支持向量机(SVM)、朴素贝叶斯(NBC)以及 $k$ 近邻(kNN)等其他分类方法对样本集做了分类测试，实验结果表 4 所示。比较实验结果可得，几种分类模型对识别 P2P

**表 2　　实验样本数目**

| 应用类型 | P2P 应用 | | | | 非 P2P 应用 | | | |
|---|---|---|---|---|---|---|---|---|
| | Thunder | PPLIVE | BT | eMule | FTP | Mail | DNS | Web |
| 训练样本数 | 90 | 90 | 80 | 120 | 120 | 50 | 80 | 60 |
| 测试样本数 | 44 | 45 | 40 | 55 | 49 | 30 | 47 | 33 |
| 总数 | 134 | 135 | 120 | 175 | 169 | 80 | 127 | 93 |

节点与非 P2P 节点的准确率均达到 96%以上，但与其他分类算法相比，决策树模型在准确度上更有一定的优势。对几种分类算法实验的漏报率和误报率分析发现，支持向量机的漏报率为 0 但是误报率较高，而决策树分类模型的误报率漏报率相对较低。因此，针对本文的 P2P 节点识别方法，决策树分类模型具有更强的适用性。

**表 4 P2P 节点的识别率**

| 分类算法 | 准确率 | 漏报率 | 误报率 |
|---|---|---|---|
| C4.5 | 338/343 (98.54%) | 4/343 | 1/343 |
| SVM | 330/343 (96.2%) | 0/343 | 13/343 |
| NBC | 332/3430(96.79%) | 9/343 | 2/343 |
| kNN | 331/343 (96.5%) | 8/343 | 4/343 |

## 5 结束语

本文通过对 P2P 应用的长期研究，根据 P2P 节点自身的特性选取了 7 个典型的特征属性，并提出了一种基于决策树模型的 P2P 节点分类方法。由于本方法只是对传输层数据分组的特征进行统计分析，因此对于加密和非加密的 P2P 应用的网络节点识别均有效。通过实验验证，该方法取得了较高的准确率和较低的漏报率及误报率。此外，由于受网络带宽、时间段分布等因素的影响，P2P 节点的特征具有一定的不稳定性，如何提高识别的准确度以及在 P2P 节点上分析出具体的 P2P 应用与其端口有待做进一步的研究。

## 参考文献：


[1] MADHUKAR A, WILLIAMSON C. A longitudinal study of P2P traffic classification[A]. Proceedings of the 14th IEEE International Symposium on Modeling, Analysis, and Simulation[C]. 2006.179-188.

[2] SEN S, SPATSCHECK O. Accurate, scalable in-network identification of P2P traffic using application signatures[A]. Proceedings of the 13th International Conference on World Wide Web[C]. 2004.512-521.

[3] LIU X B, YANG J H, XIE G G. Automated mining of packet signatures for traffic identification at application layer with apriori algorithm[J]. Journal on Communications, 2008, 29(12):51-59.

[4] CONSTANTINOUS F, MAVROMMATIS P. Identifying known and unknown peer-to-peer traffic transport layer identification of P2P traffic[A]. Proceedings of the Fifth IEEE International Symposium on Network Computing and Applications[C]. 2006.93-102.

[5] WANG J S, ZHANG Y. Connection pattern-based P2P application identification characteristic[A]. Proceedings of the 2007 IFIP International Conference on Network and Parallel Computing Workshops[C]. 2007. 437-441.

[6] RAFFAELE B, MARCO C. Characterizing the network behavior of P2P traffic[A]. Proceedings of the 2008 4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks[C]. 2008. 14-19.

[7] ALSHAMMARI R, ZINCIR-HEYWOOD A N. Can encrypted traffic be identified without port numbers, IP addresses and payload inspection? [J]. Computer Networks, 2010, 6(55):1326-1350.

[8] DUSI M, ESTE A. Using GMM and SVM-based techniques for the classification of SSH-encrypted traffic[A]. Proceedings of the 2009 IEEE International Conference on Communications[C]. 2009. 702-707.

[9] 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]. 2005. 50-60.

[10] YUAN R, LI Z, An SVM-based machine learning method for accurate internet traffic classification[J]. Information Systems Frontiers, 2012, 12(2):149-156.

[11] JUN L, SHUNYI Z, Active P2P traffic identification technique[A]. Proceedings of the 2007 International Conference on Computational Intelligence and Security[C]. 2007. 37-41.

[12] LIU F, LI Z T. Research on Server Role Based P2P Nodes Identification Methods[D]. Wuhan:Huazhong University, 2010.

[13] QUINLAN J R. C4.5 Programs for Machine Learning[M]. Morgan Kaufmann Publishers, 1993.

[14] LU G, ZHANG H L. P2P traffic identification[J]. Journal of Software, 2011, 22(6):1281-1298.

[15] LU X, DUAN H, Identification of P2P traffic based on the content redistribution characteristic[A]. Proceedings of 2007 International Symposium on Communications and Information Technologies[C]. 2007. 596-601.

[16] L7-filter[EB/OL]. http://l7-filter.sourceforge.net/, 2012.


## 作者简介：


**董永苹**（1973-），女，黑龙江建三江人，哈尔滨工业大学高级工程师，主要研究方向为计算机网络、网络安全等。

**余翔湛**（1973-），男，黑龙江哈尔滨人，博士，哈尔滨工业大学副教授，主要研究方向为网络安全、物联网安全等。

**吴刚**（1976-），男，黑龙江大庆人，哈尔滨工业大学高级工程师，主要研究方向为网络安全。