基于最短划分距离的网络流量决策树分类方法

杨哲[1,2]，**李领治**[1,2]，**纪其进**[1,2]，**朱艳琴**[1,2]

（1. 苏州大学 计算机科学与技术学院, 江苏 苏州 215006; 2. 江苏省计算机信息处理技术重点实验室, 江苏 苏州 215006）

摘 要：对不同类别的应用数据流，根据其在最初若干分组中进行握手和参数协商的差异性，通过通信模式、载荷长度以及信息熵等特征，采用基于最短划分距离的方法构建决策树模型，对其进行流量分类。经过在 4 个不同类型的真实网络数据集上的离线分类实验，以及在校园网环境中的在线流量分类实验。结果表明该模型对 8 种常见协议的网络流量，分析其前 4 到 6 个分组的特征，能够在分类准确性和系统开销上取得较好的效果。与其他机器学习算法相比，该模型构建的决策树规模较小，分类时间较短，适合于实时流量分类问题。

关键词：流量分类；双向流；最短划分距离；决策树

中图分类号：TP393 **文献标识码**：A **文章编号**：1000-436X(2012)03-0090-13

Network traffic classification using decision tree based on minimum partition distance

YANG Zhe[1,2], LI Ling-zhi[1,2], JI Qi-jin[1,2], ZHU Yan-qin[1,2]

(1. School of Computer Science and Technology, Soochow University, Suzhou 215006, China

2. Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou 215006, China)

Abstract: Before data communications, every application protocol to handshake at application layer and transmit some parameters. This process is quite different according to the protocols, such as the packet direction, payload size and the information entropy of each packet payload. So according to these features, decision tree algorithm based on minimum partition distance was used to train the classifier. The results of the offline experiments on real network traces and the online classification experiments in campus network indicate that, analyzing the first four or six packets of each flow is enough to classify eight common used application protocols with high overall accuracy and low cost. Contrast to other machine learning algorithms, decision tree can achieve better ent traces and low classification time. So it is very suitable for real-time traffic classification.

Key words: traffic classification; bidirectional flow; minimum partition distance; decision tree

1 引言

随着互联网的不断发展，出现了文件共享、VoIP 和流媒体等各种不同类型的应用流量，它们对 QoS 的要求各不相同。为了实施有效的 QoS 保障，网络管理者需要了解网络中各种类型的流量状况，关注其流量特征，或者分析相应的用户行为和应用发展趋势，从而对网络进行扩容改造。这些都必须对各种应用流量进行准确的分类。此外，准确的流量分类在网络安全、流量计费等领

收稿日期：2011-02-10；**修回日期**：2011-06-10

基金项目：国家自然科学基金资助项目（61070170）；江苏省自然科学基金资助项目（BK2009589）；江苏省高校自然科学基金资助项目（09KJD520009, 11KJB520017）

Foundation Items: The National Natural Foundation of China (61070170); The Natural Foundation of Jiangsu Province (BK2009589); The High School Natural Science Foundation of Jiangsu Province (09KJD520009, 11KJB520017)

域，也具有极其重要的意义，一直是网络测量领域的研究热点。

最初的流量分类是根据 IANA 端口映射表，将符合某个特定端口号的流量识别为相应的网络应用。其针对传统 Internet 应用和早期采用固定端口传输的 P2P 应用，分类准确性较高。随着新型应用采用随机端口、端口跳变或端口伪装等技术，该方法的分类准确性已大大下降[1]。随后 Moore[1]、Sen[2]和 Karagiannis[3]等分别提出载荷分析法，通过检查数据分组的应用层载荷，进行应用协议的特征串匹配。这些特征串称为应用层签名（application signature）。该方法分类准确性最高，为当今大多数商用系统所采用。但是其必须针对每一种应用构建应用层签名，且需经常更新以适应应用的不断发展。虽然 Haffner[4]和 Ma[5]等提出了自动构建应用层签名的方法，但只能针对传统 Internet 应用。随着应用载荷加密和混淆技术的不断发展，该方法的有效性正逐步下降。此外，也有研究者从其他的角度提出了不同的流量分类模型。Karagiannis 等[6]提出基于传输层行为的盲分类器 BLINC，通过主机在社会层、网络层和应用层 3 个层次的内在行为特性，先识别主机参与的应用，然后对其产生的流量进行分类。Xu 等[7]也采用这种办法进行流量分类，但其使用信息论和数据挖掘工具。这些方法不依赖于分组载荷，扩展性好，但模型较为复杂。Dahmouni 等[8]认为不同应用的 TCP 控制分组也是统计可分的，并提出建立 TCP 标记序列的马尔可夫链作为分类器，该方法的有效性有待进一步验证。

为了克服上述方法的不足，研究者开始寻求新的方向。一般来说，不同应用产生的流量往往具有不同的统计特征，反映着各个应用的内在特性。例如，FTP 流量的分组长度较大且分组间隔时间较小。但是，网络流量具有数据庞大、应用特性高度动态的特点。因此，利用机器学习方法，构造基于流统计特征的网络流量分类模型，是目前流量分类的研究热点。

2　基于流统计特征的网络流量分类

一般来说，流量分类的基本处理单元，通常是一组具有相同五元组<源 IP、目的 IP、源端口、目的端口、传输层协议>的分组序列，即网络流（flow）。在实际处理中，又进一步划分为单向流和双向流。单向流是指严格按照五元组规则划分的分组序列，双向流是指同一网络连接之内的双向分组序列，其包括正向流（client to server）和反向流（server to client）。通过提取单向流或双向流的特征集合，将流抽象为由一组特征属性构成的特征向量，实现从网络流量分类问题到机器学习问题的过渡。因此，利用机器学习方法处理流量分类问题，需要解决的关键问题，主要包括 3 个方面：① 选择合适的特征属性构建特征向量；② 选择适当的机器学习算法构建分类模型；③ 利用真实流量数据，对分类模型进行评价。

目前，主要使用的特征属性分为以下 3 类。

1) 数量相关特征，例如双向分组/字节总数、正向分组/字节总数和反向分组/字节总数等。

2) 长度相关特征，例如正向（反向）分组的平均长度、方差和极值等。

3) 时间相关特征，例如流持续时间、正（反）向分组的平均到达间隔、方差和极值等。

不同网络应用在这些特征属性上，一般存在较大差异。如 P2P 是双向数据传输，而被动 FTP 是单向数据传输。因此利用正（反）向分组数量的差异，可以有效区分这两者[9]。选取最能够反应网络应用本质区别的特征属性，对于网络流量分类非常重要。但特征属性之间存在相关性和冗余性，不仅增加计算复杂度，并且会降低分类准确性。Moore 等[10]提出的一个 248 项特征属性的集合，其中有 100 多项是通过傅里叶变换得来的。如果对每条网络流都进行傅里叶变换，则计算负担过于沉重。较少的特征属性可以有效降低分类模型的计算开销。Kim 等[11]经过对几种常用的机器学习算法分析后认为，使用 5~10 个特征属性进行流量分类较为合适。有些特征属性值，必须等待流结束后才能获得，例如流持续时间、分组总数等。这一限制会影响分类模型的实时性。Nguyen 等[12]提出的多子流模型，摆脱了这一限制。但子流持续时间相对较短，其特征属性容易受到网络运行状态的影响而发生变化。此外，Bernaille[13,14]、Li[15]、Huang[16]等都是根据流最初几个分组的大小和方向进行分类。这些方法具备了一定的实时检测能力，但其过分依赖于数据分组的到达顺序。在实际网络环境中由于非对称路由、网络拥塞等原因，分组通常无法保证顺序到达。因此，其稳定性得不到保证。

在获取合适的流特征属性集合之后，需要利用

机器学习方法来构建流量分类模型，并以此模型对未知类型的网络流进行分类。

Roughan 等[17]用 k-NN 和线性判别式分析方法来处理流量分类问题。Moore 等[18]采用基于概率模型的朴素贝叶斯方法，虽然该方法的整体准确率只有 65%左右，但采用基于关联的快速过滤机制 FCBF 和核估计方法，对特征集进行筛选之后，能将准确率提高到 90%以上。其他研究者还使用了贝叶斯人工神经网络[19]、支持向量机[20,21]、决策树[22,23]等有监督的机器学习方法，都取得了较好的分类准确率。Williams 等[24]分析了包括朴素贝叶斯、决策树、贝叶斯网络和朴素贝叶斯树在内的几种有监督的机器学习方法后发现，它们分类的精度大致相当，但计算开销却差别很大。之后 Kim 等人[11]综合比较了 7 种常用的机器学习方法（朴素贝叶斯、带核估计的朴素贝叶斯、贝叶斯网络、C4.5 决策树、k-NN、神经网络、支持向量机）后发现，它们都能取得 80%以上的分类准确性。2009 年，Li 等人[39]分别用 2003 年、2004 年和 2006 年的网络流量数据，对 C4.5 决策树和带核估计的朴素贝叶斯的分类准确性进行了分析。结果表明，经 2003 年的流量数据训练得到的分类模型，在对 2004 年和 2006 年的流量数据进行分类时，准确率会从 96%下降至 88%。这表明由机器学习得到的分类模型具有时效性。2010 年，Callado 等[40]进一步分析了载荷分析法和 6 种机器学习方法（NBTree、PART、J48、贝叶斯网络、带核估计的朴素贝叶斯和支持向量机），在校园网及商用网络中对流量分类的准确性。其结果表明在校园网环境的网络条件比较均衡，应用种类较少，机器学习算法能够取得 95%的准确性。而在商用网络中，网络环境较为复杂，应用种类较多，且存在一定的未知应用，导致机器学习算法的分类准确性均低于 77%，贝叶斯网络方法的准确性甚至只有 10%左右。

但是，有监督的机器学习方法，都需要用预先分类好的样本集来训练分类器。如果分类样本较少，会出现过度拟合（overfitting），导致分类器的泛化能力弱。因此，有些研究者用无监督的机器学习方法进行流量分类。McGregor 等[25]最早使用 EM 算法对网络流量进行聚类分析。Erman 等[26,27]采用了 K-Mean、DBSCAN 和 AutoClass 算法来进行流量聚类，其精度明显优于朴素贝叶斯。Yuan 等[28]分析了端口、IP 地址、字节数等特征的熵，用聚类方法进行流量分类。之后 Erman 等[29]又提出了半监督的学习方法，有效减少了预处理时间，并且能够达到较高的分类精度。此类方法在聚类过程中无须使用训练样本的类型，因而能够识别部分类型尚未定义的新型网络流量。然而在聚类结束后必须进行手工标记以实现网络流量的分类，效率偏低。

从目前的研究成果来看，多数分类模型所采用的特征属性，必须等待流结束后才能获取。这大大影响了分类的实时性和实用性。虽然也有研究者仅仅根据流最初几个分组的大小和方向进行分类，使得对流量的分类有可能在流刚开始的一段时间内实现，但其过分依赖于数据分组的到达顺序，稳定性得不到保证。因此，本文对不同类别的应用数据流，根据其在最初若干分组中进行握手和参数协商的差异性，通过通信模式、载荷长度以及信息熵等特征，采用基于最短划分距离的方法构建决策树模型，对其进行流量分类。通过在 4 个不同类型的网络数据集上的离线分类实验，以及在校园网环境中在线流量分类实验，结果表明本方法对 8 种常见的网络应用，能够在分类准确性和系统开销上取得较好的综合效果。

3 双向流特征属性

本文进行分类处理的对象是双向流，因其不仅包含通信双方在 2 个方向上独立的网络流特征，还包含 2 个单向流之间的关联特征，具有更强的特征描述能力。由于 UDP 协议是面向无连接的，与大多数分类模型一样，本文暂不考虑。对于 TCP 协议，由于双向 TCP 流是一个双向有序的分组序列，为统一对方向问题的讨论，本文以双向 TCP 流的 SYN 分组的方向为该流的正方向。在计算流特征属性时，忽略状态控制分组（如 SYN、RST 和 FIN 等）。因为这些分组的应用层载荷长度一般为零，对流量分类的作用不大。

在流量分类问题研究中，分类的类别可以有不同的定义。如将一组功能相似的应用定义为同一类，也可深入识别确切的应用甚至软件版本。本文将分类的类别定义为具体的应用协议，具体包括 HTTP、HTTPS、FTP、SMTP、POP3、IMAP、BitTorrent、eDonkey 在内的 8 个典型 Internet 应用协议。

对于任何一种网络应用，通信双方会按协议状

态机，在某个时刻 i 发出包含不同载荷数据的分组 X_i。随着时间的推移，通信双方之间一次完整的流会话过程，可以视为一个离散平稳信源，用随机变量序列 $X=X_1X_2X_3X_4\cdots X_i\cdots$表示。待分类的网络流对象 X 可以由若干特征属性描述，设计通用的分类模型时应避免选择基于特定协议的特征。对于不同的应用，2 个方向上的流特性迥异，还应考虑分别选取 2 个方向上的特征属性。出于实时性方面的考虑，所选取的特征属性，应该能通过流早期的若干分组计算得出。

一般来说，在消息序列 X 的前 N 个消息中，通信双方需要进行应用层握手，并协商各种参数。根据不同应用协议的约定，需要协商和传递的参数个数和类型不同，会导致其最初若干消息分组的方向、载荷长度以及载荷中每个字节取值的不确定性和差异性，其最能反映出应用协议的本质特征。因此本文使用以下特征，用于刻画前 N 个消息分组的这种差异性。

3.1　通信模式

通信模式（pattern），用于描述双向流中前 N 个分组的方向，用形如 $b_1b_2b_3\cdots b_N$ 形式的二进制整数表示，其中，b_i（$i\leqslant N$）用于表示第 i 个分组的方向。本文规定如果第 i 个分组的方向与流量的正向同向，则 b_i 取值为 1，否则为 0。例如 $N=4$ 时，则 Pattern 的取值为 0000～1111。

如图 1 所示的 FTP 应用，在 TCP 握手之后登录过程中，第 1 个分组方向是由服务器端发往客户端的“220”消息(b_1=0)，之后 3 个分组的方向交替出现。因此，FTP 应用的通信模式一般为 0101。而 HTTP 的第一个分组是由客户端发往服务器端的 Get 请求(b_1=1)，之后 3 个分组的方向受请求和响应报文的长度影响，没有固定的模式，因此 HTTP 的通信模式可能为 1000～1111。

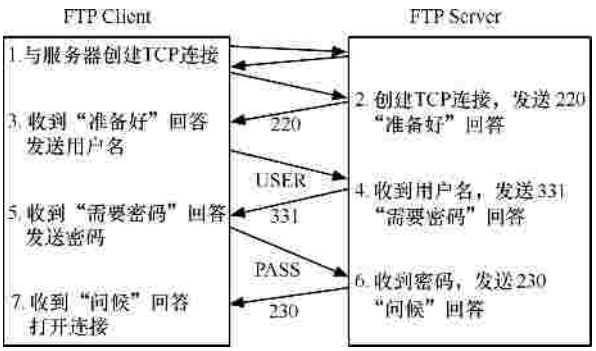

图 1　FTP 登录过程的通信模式（$b_1b_2b_3b_4$=0101）

一般来说，不同应用协议的通信模式是统计可分的。本文对 WIDE 数据集（详见 5.1 节）中的 FTP、HTTP、BitTorrent 和 eDonkey 应用的通信模式进行统计，发现其通信模式有明显的差异，如表 1 所示（单位为%）。FTP 和 HTTP 是典型的 C/S 应用，有 92.84%的 FTP 流的通信模式为 0101，有 74.39%的 HTTP 流，其通信模式为 1000。而 BitTorrent 和 eDonkey 这 2 种典型的 P2P 应用，其通信模式主要为 1010 和 1001。

表 1　FTP/HTTP/BitTorrent/eDonkey 协议的通信模式

通信模式 $b_1b_2b_3b_4$	协议类型			
	Client/Server		P2P	
	FTP	HTTP	BitTorrent	eDonkey
0000	2.19	0.00	0.00	0.00
0001	1.19	0.00	0.00	0.00
0010	3.43	0.00	0.00	0.00
0011	0.00	0.00	0.01	0.00
0100	0.33	0.00	0.00	0.00
0101	**92.84**	0.00	0.01	0.00
0110	0.02	0.00	0.01	0.00
0111	0.00	0.00	0.00	0.00
1000	0.00	**74.39**	0.73	0.20
1001	0.00	3.34	6.09	**84.41**
1010	0.00	15.79	**83.73**	14.13
1011	0.00	0.17	6.94	0.04
1100	0.00	3.44	0.07	0.47
1101	0.00	1.06	1.47	0.74
1110	0.00	1.50	0.08	0.01
1111	0.00	0.32	0.86	0.00

3.2　载荷长度

双向流消息序列中，任一时刻发出的消息 X_i，其载荷长度（payload size）因协议和传输的参数而异。因此，本文用 S_i 表示消息分组 X_i 的应用层载荷大小。与其他研究者使用分组大小[12,13,15~18,22]特征不同的是，本文使用的是以字节为单位的应用层载荷大小。因为分组中可能含有 IP 或 TCP 的可选数据，使用分组大小不能准确描述各种应用协议的应用层握手或信令长度的特征。

3.3　分组信息熵

对任一消息 X_i，其载荷内容受协议和传输参数而异，其每个字节都取且可取遍字符集 $K=\{k, 0\leqslant k$

≤255}。本文将分组 X_i 载荷数据中每个字节的取值当作一组随机事件，即 $X_i=\{x_k, 0\leqslant k\leqslant 255\}$，用 x_k 表示在分组 X_i 的载荷数据中，值为 k 的字节出现的次数，则 X_i 的信息熵为

$$H(X_i)=-\sum_{k=1}^{255}p(x_k)\mathrm{lb}p(x_k) \tag{1}$$

其中，

$$p(x_k)=x_k\Big/\sum_{k=1}^{255}x_k=x_k/S_i \tag{2}$$

对于一个完整的双向流消息序列 X，其前 N 个分组的信息熵构成了一个熵序列 $H(X_1)$，$H(X_2)$，…，$H(X_N)$，反映了每个分组 X_i 各自所含信息量的多少。但双向流是一个双向有序的分组序列，而由于 $H(X_i)$ 的非负性（$H(X_i)\geqslant 0$），分组信息熵只能刻画信息量的多少，不能描述分组的方向。因此本文定义分组 X_i 的有向熵（directed entropy）$\vec{H}(X_i)$，用于增强对方向性的描述，$\vec{H}(X_i)$ 可定义为

$$\vec{H}(X_i)=\mathrm{sgn}(b_i-0.5)H(X_i) \tag{3}$$

其中，sgn(*x*)为符号函数，b_i 为 Pattern 值，表示分组 X_i 的方向，其取值为 1 或 0。若 X_i 与流的正向同向，则 $b_i=1$，$\mathrm{sgn}(b_i-0.5)=1$，$\vec{H}(X_i)=H(X_i)\geqslant 0$，表示此时双向流按正向传递了大小为 $H(X_i)$ 的信息。若 X_i 与流的正向相反，则 $b_i=0$，$\mathrm{sgn}(b_i-0.5)=-1$，$\vec{H}(X_i)=-H(X_i)\leqslant 0$，表示双向流按负向传递了大小为 $H(X_i)$ 的信息。此时，双向流前 N 个分组的有向熵构成的熵序列 $\vec{H}(X_1)$，$\vec{H}(X_2)$,…,$\vec{H}(X_N)$，从微观上能有效刻画双向流的信息传递大小和方向。此外，为了从整体上描述双向流的信息流动情况，本文定义了双向流 X 的有向信息熵 $\vec{H}(X)$：

$$\vec{H}(X)=\sum_{i=1}^{N}\vec{H}(X_i)=\sum_{i=1}^{N}\mathrm{sgn}(b_i-0.5)H(X_i) \tag{4}$$

对于多数 P2P 流，由于其双向数据传递的特点，其 $|\vec{H}(X)|$ 值一般较小，甚至趋近于 0。而对于采用 C/S 模式的应用协议，一般其 $|\vec{H}(X)|$ 较大。因此，双向流的有向信息熵能从宏观上有效区分 P2P 和 C/S 模式的应用数据流。

3.4 双向流特征属性集

因此本文使用表 2 所示的特征属性作为双向流的分类依据。虽然端口号已经不能作为主要的分类特征，但它仍是一个重要特征，尤其是对一些传统的 Internet 应用进行分类时[2]。因此，本文仍使用端口号（port）作为描述双向流的特征之一。

表 2 双向流的特征属性

名称	描述
port	正整数，双向流 *X* 使用的传输层端口
Pattern($b_1b_2b_3\cdots b_N$)	正整数，双向流 *X* 的通信模式
$S_1\sim S_N$	正整数，分别为前 *N* 个分组的载荷大小
$\vec{H}(X_1)\sim\vec{H}(X_N)$	正实数，分别为前 *N* 个分组的有向信息熵
$\vec{H}(X)$	正实数，为双向流 *X* 的有向信息熵

4 基于最短划分距离的决策树

本文上述特征属性，采用决策树算法对真实网络流量进行分类。决策树是以实例为基础的归纳学习算法，能从一组无序的已知样本中，归纳出以倒挂的树状结构表示的分类知识。决策树的每个内部节点，代表对一个或多个特征属性的取值进行测试比较，并根据比较结果确定该节点的分枝，每个叶节点就代表一个类别。决策树算法应用简便，只要训练样本集能够用属性向量和类别表示，就能使用该算法。在使用决策树模型对类型未知的样本进行分类时，只需从根节点开始逐步对该样本的特征属性进行测试，并沿着相应的分支直到某个叶节点为止，此时叶节点所代表的类型即为该样本的类型。与其他机器学习算法相比，决策树算法相对简单，具有较高的数据处理效率，适合进行流量的实时分类[22,23]。

4.1 基于信息熵的决策树

在决策树创建过程中，如何确定内部节点的分枝是最为关键的问题。对当前节点进行分枝，对应着当前样本集的一个划分。采用不同的划分标准得到的决策树各不相同。在网络流量分类问题中，T 为已知类别的流量样本数据集，包含 m 种类别 $\{c_1,c_2,\mathrm{L},c_m\}$，每个流样本的特征属性集为 $A=\{a_1,a_2,\mathrm{L},a_k\}$。根据样本类别可得 T 的一个理想划分 $T=\{t_1,t_2,\mathrm{L},t_m\}$，其中，样本子集 t_i 中的样本都属于类别 c_i，则该理想划分的信息熵为

$$H(T)=-\sum_{i=1}^{m}p(t_i)\mathrm{lb}p(t_i) \tag{5}$$

$$p(t_i)=|t_i|\Big/\sum_{i=1}^{m}|t_i| \tag{6}$$

如果以选择特征属性 $a_i\,(1\leqslant i\leqslant k)$ 作为测试条

件 a，根据该属性的不同取值，可得 T 的另一个划分 $T'=\{t_1',t_2',\L,t_r'\}$，则该划分的信息熵为

$$\begin{aligned}H_a(T')&=\sum_{i=1}^{r}p(t_i')H(t_i')\\&=\sum_{i=1}^{r}p(t_i')\left(-\sum_{j=1}^{m}p(t_j\mid t_i')\mathrm{lb}p(t_j\mid t_i')\right)\\&=-\sum_{i=1}^{r}\sum_{j=1}^{m}p(t_i')p(t_j\mid t_i')\mathrm{lb}p(t_j\mid t_i')\end{aligned}\tag{7}$$

其中，

$$p(t_i')=|t_i'|\Big/\sum_{i=1}^{r}|t_i'|\tag{8}$$

$$p(t_j\mid t_i')=|t_i'\cap t_j|\Big/|t_i'|\tag{9}$$

式(9)表示划分 T' 中属于子集 t_i' 的样本，在理想划分中属于子集 t_j 的概率。因此，选择 a_i 对 T 进行划分所获得的信息熵增益为

$$H_{\mathrm{Gain}}(T,a)=H(T)-H_a(T')\tag{10}$$

ID3 算法中[30]，根据每个测试条件的信息熵增益，选择增益最大的测试来确定当前节点的分枝。对大多数应用，信息熵增益法都能取得较好的结果。但是其最大的缺陷是偏爱输出分枝较多的测试条件，导致决策树规模过大，泛化能力弱。随后在改进的 C4.5 算法中[31]，引入测试条件 a 带来的分割信息熵。

$$H_{\mathrm{Split}}(T,a)=-\sum_{i=1}^{r}p(t_i')\mathrm{lb}p(t_i')\tag{11}$$

用信息增益与分割信息熵的比值，即信息增益率作为选择测试条件的依据。

$$Ratio(T,a)=H_{\mathrm{Gain}}(T,a)\Big/H_{\mathrm{Split}}(T,a)\tag{12}$$

但是采用增益率标准，当划分产生的分割信息量很小时，增益率的值不稳定。一种情况是，如果划分后只有一个子集中有样本，会导致分母——分割信息熵为零。另一种情况是，对于某些划分，虽然产生的信息熵增益很小，但由于分割信息熵也很小，其信息增益率很大。

4.2　基于划分距离的决策树

为克服 C4.5 算法存在的上述 2 个缺点，本文用 Mantaras 范式距离来定义 2 个划分间的距离[32]。在各属性中选择与理想划分距离最近的属性作为当前节点的测试条件，用最短距离划分的办法来构建决策树。

根据样本集 T 的理想划分 $T=\{t_1,t_2,\L,t_m\}$ 和以属性 a_i 作为测试条件 a 所得的划分 $T'=\{t_1',t_2',\L,t_r'\}$，则划分 T' 对于理想划分 T 的条件熵为

$$H(T\mid T')=-\sum_{j=1}^{m}\sum_{i=1}^{r}p(t_i',t_j)\mathrm{lb}\left(p(t_i',t_j)\Big/p(t_i')\right)\tag{13}$$

其中，

$$p(t_i',t_j)=|t_i'\cap t_j|\Big/\sum_{i=1}^{r}|t_i'|=|t_i'\cap t_j|\Big/\sum_{j=1}^{m}|t_j|\tag{14}$$

式(14)表示一个样本在划分 T' 中属于子集 t_i'，而在理想划分 T 中属于子集 t_j 的概率。同理，理想划分 T 对于划分 T' 的条件熵为

$$H(T'\mid T)=-\sum_{j=1}^{m}\sum_{i=1}^{r}p(t_i',t_j)\mathrm{lb}\left(p(t_i',t_j)\Big/p(t_j)\right)\tag{15}$$

划分 T' 与理想划分 T 的联合熵为

$$H(T',T)=-\sum_{j=1}^{m}\sum_{i=1}^{r}p(t_i',t_j)\mathrm{lb}p(t_i',t_j)\tag{16}$$

因此，划分 T' 与理想划分 T 的 Mantaras 范式距离为

$$d(T',T)=\frac{H(T'\mid T)+H(T\mid T')}{H(T',T)}\tag{17}$$

按照最短距离标准，首先计算每个可能的特征属性所对应的划分与理想划分之间的距离，然后选择距离最小的划分所对应的属性作为当前节点的测试属性。最短距离法能有效克服信息增益率方法的缺陷，算法的稳定性好，其构建的决策树规模更小，分类速度更快。

4.3　属性离散化和剪枝

如表 2 所示，本文使用的流特征中，除端口和通信模式为离散属性外，其余均为连续属性。如果直接处理连续属性，一旦被选为决策树内部节点的测试条件，则会产生很多分枝，造成决策树结构庞大。因此，必须对连续属性进行离散化。本文采用划分距离评估连续区间的分割点，即选择划分后的 2 个样本子集间 Mantaras 距离最小的分割点作为最佳分割点[32]，以最短描述长度准则（MDLP，minimum description length principle）作为离散化过程的结束条件。

在无冲突的情况下，算法生成的决策树将与训练集完全一致。但由于训练集里的实例可能包含了噪声点和孤立点，训练中生成的树会尝试拟合每一个异常的细节。这种过度拟合的结果，是对训练集分类的效果很好，但用它来对新的数据集进行分类时可能并不理想。因此有必要对生成的初始决策树进行剪枝，以得到更一般的分类规则。因此，本文

使用 PEP（pessimistic error pruning）算法[31]对生成的初始决策树进行剪枝，进而得到最终的决策树。由于 PEP 算法在剪枝过程中，对每棵子树最多只检查一次，且不需要额外的剪枝数据集，因此其速度较快，适用于样本数较多的数据集。

此外，在初始特征属性集中，属性之间存在的相关性和冗余性，不仅增加计算复杂度，并且很可能会降低分类的准确性。一般可以利用特征过滤算法，发现属性之间的关联性，从而滤除冗余属性。但是特征过滤算法可能会利用局部信息过滤特征属性，造成局部最优性问题。而且在决策树本身的构造中，实际上已经包含了属性选择过程，所以本文没有采用特征过滤算法。

5 实验与分析

5.1 实验环境

为了验证本文方法在分类准确性和实时性方面的性能，实现了一个原型系统，并部署在苏州大学计算机科学与技术学院局域网边缘的路由交换机上，其拓扑结构如图 2 所示。

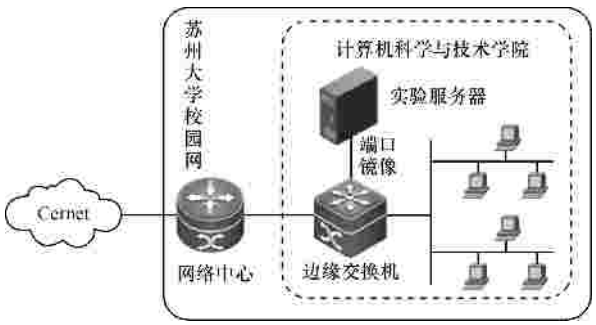

图 2 实验环境拓扑结构

该网络拥有约 1 200 台主机，其边缘为 Cisco 4507 路由交换机，通过吉比特每秒链路与苏州大学网络中心核心路由器相连。在该边缘路由交换机上通过端口镜像，将其出口的全部流量镜像到本文实验用的 Dell PowerEdge 1950 服务器上。服务器配置为 1 路 4 核至强 CPU，8GB 内存，146GB 硬盘，运行 Windows 7 操作系统。该服务器主要用于：① 用真实网络的离线流量数据，训练基于最短划分距离的决策树模型，并与其他分类算法的准确性进行对比；② 将经过训练得到的分类模型，用于在线流量分类，以验证其实时分类的能力。

5.2 网络流量数据与工具

本文首先使用 4 个真实网络流量数据，对决策树分类模型进行离线训练和验证。数据的采集点包括局域网和广域网，时间跨度从 2003 年到 2011 年，分别是 LBNL[33]、UNIBS[34]、WIDE[35]和 SUDA 数据集。

LBNL 数据集来自劳伦斯伯克利国家实验室（lawrence Berkeley national laboratory）局域网出口的 Trace 数据集，本文选用 2003 年 1 月 12 日的数据。UNIBS 数据集来自 University of Brescia 办公网，约有 1 000 个左右的用户，通过吉比特以太网链路接入 Internet。采集时间为 2007 年某一周的白天。WIDE 数据集来自 WIDE 互联网测量与分析项目采集的骨干网链路的 Trace 数据。本文采用的是 F 点获取的 2009 年 10 月 1 日 20:00～21:00 的数据。其中，LBNL 和 UNIBS 原始 Trace 中包含全部分组载荷，WIDE 原始 Trace 中，每个分组包含 40byte 的载荷内容，三者对涉及用户隐私的信息经过匿名处理。最后的 SUDA 数据集，采集自 5.1 节所述的苏州大学计算机科学与技术学院的局域网出口，采集时间为 2011 年 4 月 13 日至 15 日，数据集中每个分组包含 40byte 的载荷。

对上述 4 个离线数据集，首先使用 Tcptrace 软件[36]提取出双向均发送一个以上分组的双向 TCP 流，并计算表 2 所述双向流的特征属性。本文分类的对象为 HTTP、HTTPS、FTP、SMTP、POP3、IMAP、BitTorrent、eDonkey 等 8 种应用协议。首先使用 Analyzer 载荷分析软件[38]，对离线数据集中这 8 种应用的双向 TCP 流，通过其应用层签名进行标记。经过流特征属性提取和标记后的数据集中，各种应用协议双向流的数量如表 3 所示。其中，LBNL 数据集中没有 BitTorrent 和 eDonkey 的流量，UNIBS 数据集中没有 IMAP4 的流量。

表 3 数据集中各种协议的样本数

协议名	数据集			
	LBNL	UNIBS	WIDE	SUDA
HTTP	81 984	7 063	307 799	12 478
HTTPS	18 013	25 427	362 858	13 589
SMTP	20 825	19 427	12 949	30 458
POP3	1 172	19 611	8 974	35 789
IMAP4	7 677	/	5 879	1 487
FTP	22 178	6 296	204 576	4 789
BitTorrent	/	5 057	106 975	94 363
eDonkey	/	1 578	70 291	47 567
合计	151 849	84 459	1 080 301	240 520

本文的决策树分类模型及其剪枝算法，是基于 Weka 平台[37]实现的。用于与本文方法进行对比的

常用分类算法有 k-NN（k-nearest neighbors）、SVM（support vector machines）、NBK（naïve bayes kernel estimation）和 C4.5 决策树算法，这些算法也都基于 Weka 平台实现。

5.3　评价指标

本文采用如下准确性指标对分类模型的分类能力进行评估。在训练数据集中的流样本，分别属于 m 种类别。对类别 i，令 TP_i 表示其全部样本中被正确分类的样本数，FN_i 表示被误判为其他类别的样本数，FP_i 表示其他类别的样本中被误判为类别 i 的样本数，则定义如下分类准确性指标。

1) 类别 i 的召回率（recall）

$$R_i = TP_i / (TP_i + FN_i) \tag{18}$$

2) 类别 i 的准确率（precision）

$$P_i = TP_i / (TP_i + FP_i) \tag{19}$$

3) 类别 i 的 F 值（F-measure）

$$F_i = 2 \times P_i \times R_i / (P_i + R_i) \tag{20}$$

4) 整体准确率（overall accuracy）

$$OA = \sum_{i=1}^{m} TP_i \Big/ \sum_{i=1}^{m} (TP_i + FP_i) \tag{21}$$

上述准确性指标中，单个类别的召回率和准确率能够反映分类模型针对某个类型的分类能力，而 F 值对模型分类能力的综合评价能力更好，因为 F 值是召回率和准确率的调和平均值。此外，分类模型的整体准确率，能从整体上反映正确分类的样本数占总样本数的比例。因此它和 F 值指标的应用最广，几乎为所有研究所采纳[2]。另外，本文还使用训练时间、分类时间、吞吐率等指标来评估分类模型的开销。

本文使用每个流的机器处理时间来评价分类模型的实时分类性能。流处理时间是指从每个流的第一个分组进入机器开始，然后计算流特征属性，最后由分类模型对流所属应用类型进行标记的全部时间，其包括 3 个部分。

1) 流缓存时间 t_{cache}。由于本文用表 2 所定义的流特征属性，需要处理每个 TCP 流中前 N 个载荷长度不为零的分组。因此 t_{cache} 时间是指每个流的第一个分组进入机器开始，到前 N 个载荷长度不为零的分组全部到达的时间。

2) 属性计算时间 $t_{\text{attributes}}$，是指按表 2 的定义计算 TCP 流的特征属性的时间。

3) 分类时间 $t_{\text{classification}}$，是指分类模型依据每个流的特征属性，对流所属应用类型进行标记的时间。

这 3 部分机器处理时间，都与分类模型需要对每个流进行处理的分组个数 N 有关。N 越大，则 3 部分时间都会延长，导致分类模型对流所属应用类型进行标记的时机越迟，其实时性变差；而 N 越小，虽然有利于提高分类实时性，但会导致分类的准确性降低。因此，本文用这 3 个指标来评价模型的实时分类能力，同时考虑准确性指标，以获得最佳的分类效果。

5.4　离线训练和验证

本文首先在 4 个离线数据集上，进行了 3 个实验，主要对本文提出的基于最短划分距离的决策树模型的分类准确性进行评价。其中，第 1 个实验用于确定本文模型中的待定参数 N，即每个流需要分析的分组数量。第 2 个实验在确定 N 取值的条件下，对比各种机器学习方法的分类整体准确性。第 3 个实验用于分析抽样率对单个类别分类准确性的影响。

实验 1　确定待定参数 N。

为了确定每个流需要分析的分组个数 N，首先将 4 个离线数据集，用随机抽样的方法分别分成相同大小的 A、B 2 个子集，子集中每类应用协议的流样本的比例与原数据集保持一致，然后分别用不同的 N 取值，在子集 A 上进行训练以获得决策树分类模型，在子集 B 上对模型进行验证。重复实验 10 次，每次重新对离线数据集进行随机抽样，获得 10 组不同的 A、B 子集。综合考虑 10 次实验中分类模型的平均整体准确率和系统开销等指标后，确定最佳的 N 取值。

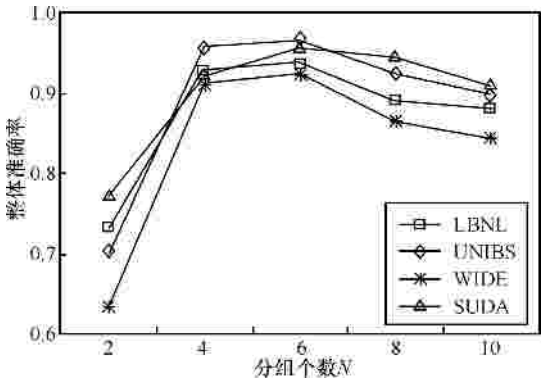

图 3　不同 N 取值下的整体准确率

图 3 给出了不同 N 取值（N＝2, 4, 6, 8, 10）条件下，分类模型在 4 个离线数据集上的整体准确率指标。可以看出，每个流如果只分析 2 个分组的信

息，因为获取的信息不充分，导致整体准确率较低。分类模型的整体准确率随着 N 值的增加而提高。当 N 取 6 时，即每个流分析的分组数量为 6 个时，分类模型的整体准确率达到最大值，之后则略有下降。因为大多数协议的应用层握手过程，一般都在 4～6 个分组中完成，少数会持续 6～8 个分组，之后便开始传输数据。进行数据传输时，受数据内容和长度的影响，其分组的方向、长度和信息熵等特征不稳定，导致分类器的整体准确率有所下降。

表 4 给出了不同 N 取值条件下，分类模型在 4 个数据集上的训练和分类时间，以及吞吐率指标。决策树分类模型的建模过程相对复杂，因此导致其训练时间较长。但分类时只需根据流的特征属性在树状模型中，自上而下的进行简单比较，因此分类时间要明显少于训练时间，适合于实时的流量分类问题。但是随着 N 的增加，需要计算和进行测试的特征属性较多，训练和分类时间都大大增加，吞吐率也逐渐下降。

因此，从分类的整体准确率考虑，N＝6 是最佳的取值。此时在 4 个数据集上的分类准确性分别为 94.0%（LBNL），96.7%（UNIBS），92.6%（WIDE）和 95.8%（SUDA），平均吞吐率保持在 5.1×10^4 流/秒。

实验 2 不同分类算法的准确性对比。

在第 1 个实验得到的 10 组 A、B 子集上，实验 2 分别用不同的机器学习算法，用本文使用的特征集进行训练和分类，以对比本文的基于最短划分距离的决策树（DBDT, distance based decision tree）算法和其他机器学习算法，在分类准确性和分类性能的差异。

图 4 给出了在 4 个离线数据集上，每种算法的分类整体准确率指标。可以看到，k-NN 和 NBK 算法的准确率较低，本文的 DBDT 算法和 SVM、C4.5 算法的整体准确率大致相当，在样本数量较少的 UNIBS 数据集上，略低于 SVM 算法。这是因为 SVM 算法是专门针对小样本情况下的机器学习算法，而决策树算法无论样本规模如何，都能取得较好的分类准确性。

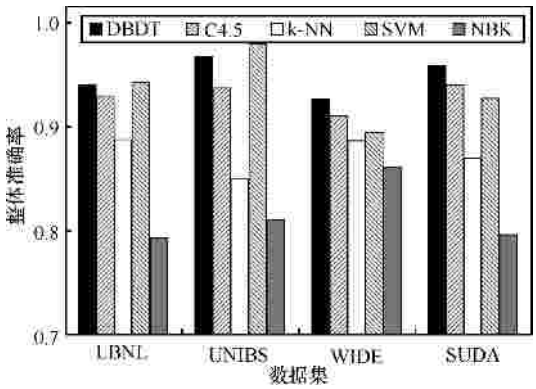

图 4 5 种算法的分类整体准确率比较

表 5 给出了在 N＝6 的条件下，5 种分类算法在 4 个数据集上训练和分类时间以及吞吐率指标。

表 4 不同 N 取值下的算法开销

N	训练时间/s				分类时间/s				吞吐率[注](flow/s)			
	LBNL	UNIBS	WIDE	SUDA	LBNL	UNIBS	WIDE	SUDA	LBNL	UNIBS	WIDE	SUDA
2	6.99	6.69	24.62	7.05	1.65	1.56	5.74	1.21	9.6×10^4	9.2×10^4	9.4×10^4	9.9×10^4
4	8.49	6.76	36.83	8.57	1.96	1.60	8.33	1.64	6.7×10^4	6.0×10^4	6.5×10^4	7.3×10^4
6	19.25	14.19	66.66	24.58	3.15	2.22	10.46	2.19	5.1×10^4	4.7×10^4	5.2×10^4	5.5×10^4
8	34.44	21.85	129.81	40.78	4.07	2.69	15.83	3.48	3.5×10^4	3.4×10^4	3.4×10^4	3.5×10^4
10	54.09	37.35	245.95	72.48	5.28	3.61	23.52	4.97	2.2×10^4	2.1×10^4	2.3×10^4	2.4×10^4

注：吞吐率＝流数量/分类时间（单位：流/秒）

表 5 5 种算法的性能对比

N	训练时间/s				分类时间/s				吞吐率(flow/s)			
	LBNL	UNIBS	WIDE	SUDA	LBNL	UNIBS	WIDE	SUDA	LBNL	UNIBS	WIDE	SUDA
DBDT	19.26	13.48	67.42	24.58	1.50	0.90	10.46	2.19	5.1×10^4	4.7×10^4	5.2×10^4	5.5×10^4
C4.5	21.45	16.25	77.47	30.44	2.07	1.45	13.16	3.63	3.7×10^4	2.9×10^4	4.1×10^4	3.3×10^4
k-NN	284.22	198.37	683.74	384.86	103.28	87.30	573.20	231.84	735	484	942	519
SVM	488.47	277.34	1063.27	553.39	47.47	25.32	289.36	67.96	1.599×10^3	1.668×10^3	1.867×10^3	1.770×10^3
NBK	10.35	7.22	67.43	14.86	5.37	3.28	35.27	9.52	1.4×10^4	1.3×10^4	1.5×10^4	1.3×10^4

从训练时间来看，决策树算法（DBDT 和 C4.5）并不是最优的。但是在分类时间和吞吐率上，决策树算法仅需根据样本的特征属性，在决策树上自上而下的依次比较，处理相对简单，因此具有较高的数据处理效率。与 C4.5 决策树算法相比，DBDT 采用最短划分距离构建决策树，其克服了 C4.5 算法的缺陷，构建的决策树规模更小，因此其训练和分类速度更快。

实验 3　抽样率对准确率的影响。

在第一个实验得到的 10 组 A、B 子集上，实验 3 主要分析不同抽样率对每个类别的分类准确率的影响。首先随机抽取 A 子集中每类应用协议 10%的流样本，进行训练以获得分类模型，然后在子集 B 上对模型进行验证。随后抽样率按 10%递增直至 100%，重复实验 10 次，以获得不同抽样率下单个类别的平均分类准确性指标。图 5~图 8 分别给出了在 LBNL、UNIBS、WIDE 和 SUDA 数据集的 A 子集上，用不同抽样率进行训练后，用 B 子集进行验证得到 *F* 值的 10 次实验平均值。

如图 5 所示，在 LBNL 数据集上，每个类别的 *F* 值随着抽样率的递增而增加。但在原始数据集中，由于 POP3 和 IMAP4 协议样本数量较少，分别为 1 172 和 7 677 个流。在对 A 子集进行小样本抽样（10%~50%）后，样本数量严重不足，导致分类器训练不充分，对某些特殊样本过度训练。因此 *F* 值较低，且波动较大。而 HTTP、SMTP 和 FTP 的样本数量相对较多，虽然在小样本抽样条件下的 *F* 值较低，但仍高于 POP3 和 IMAP4。

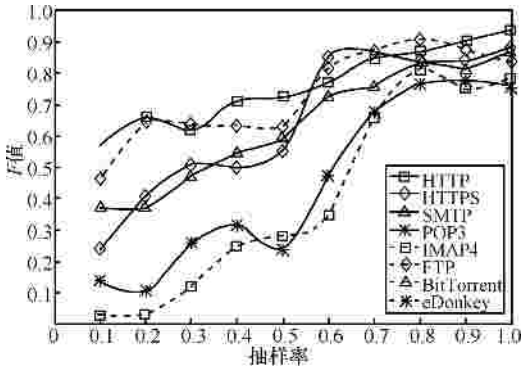

图 5　LBNL 数据集上的 *F* 值

如图 6 所示，在 UNIBS 数据集上，每个类别的 *F* 值随着抽样率变化的趋势与 LBNL 相似，即对 eDonkey 协议流的分类 *F* 值较低，HTTPS、SMTP 和 POP3 的分类 *F* 值较高。

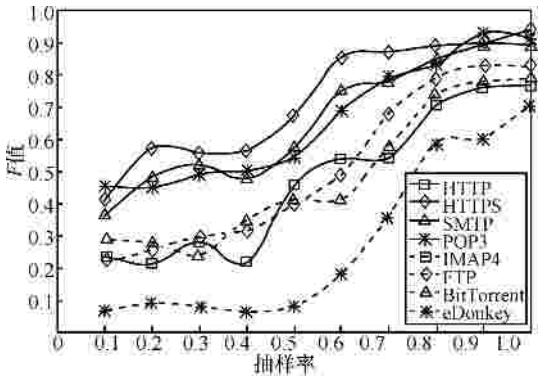

图 6　UNIBS 数据集上的 *F* 值

如图 7 所示，在 WIDE 数据集上，每个类别的 *F* 值的变化趋势与前两者相似，但整体的波动性要小。这是由于 WIDE 数据集中，每个类别的样本绝对数量要大于前两者，小样本抽样后得到的样本数量仍然较多。虽然对分类器的训练仍然不充分，导致 *F* 值较低，但在特殊样本上进行过度训练的概率较低，因此 *F* 值的波动比较平稳。

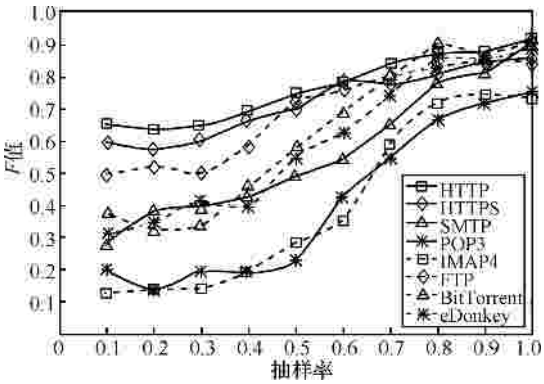

图 7　WIDE 数据集上的 *F* 值

如图 8 所示，在 SUDA 数据集上，除 IMAP4 和 FTP 外，其他类别的 *F* 值变化趋势与 WIDE 相似。这是由于 SUDA 数据集中，IMAP4 和 FTP 类别的样本较少，而其他类别的样本较多。

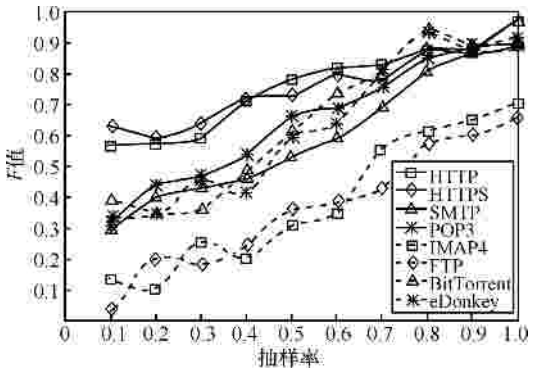

图 8　SUDA 数据集上的 *F* 值

因此，针对单个类别的应用流量，为了取得较好的分类准确性，其用于训练的流样本的绝对数量必须保持一定的规模，才能避免分类模型对噪声点和孤立点的过度拟合，导致分类准确性的波动较大。

5.5 在线流量分类

除验证本文模型的准确性指标外，在 5.1 节所述的环境中，还进行了流量的在线分类实验，以验证其实时分类能力。首先在实验服务器上，用 Tcptrace[36]捕获镜像流量中的 TCP 流，然后计算表 2 所定义的流特征属性。利用离线实验中，经由 SUDA 离线流量数据训练得到的分类模型进行流量的在线分类，用流处理时间指标来评价其实时分类的性能。同时在实验服务器上，还部署了 Analyzer 软件，用于事后检验分类的准确性。

考虑到内存开销和计算复杂度问题，实验中没有捕获每个流的所有分组。根据图 3 和表 4 所示的离线实验结果，从准确性角度考虑，$N=6$ 时是最佳结果。但是，在线流量分类对时间的要求更为关键。根据 5.2 节对机器处理时间与 N 取值的关系分析，在同时保持较高准确性和分类性能的前提下，$N=4$ 也是较好的结果。因此，在线分类实验一共持续 6 天时间，从 2011 年 4 月 17 日到 22 日。其中前 3 天的实验，N 的取值为 6，即捕获了每个双向 TCP 流的前 6 个载荷不为零的分组；后 3 天的实验中，N 取 4，用于验证 N 取值与模型实时分类能力的影响。另外，考虑到实际网络中存在一定比例的短流（有效分组数少于 N 个）以及其他 TCP 错误，因此实验中对每个流的处理设置一个超时时间 $t_{\text{timeout}}=10\text{s}$。如果在 t_{timeout} 时间内，一个 TCP 流的有效分组数量少于 N，则不进行处理，以节约内存的开销。

本文首先分析了不同的 N 取值，对在线分类的整体准确性的影响，如图 9 所示。实验每 6 个小时统计一次在线分类的整体准确率指标，在 N 分别取 6 和 4 的各自 3 天实验中，一共取得 12 组数据。通过对比可以看到 $N=6$ 的 3 天实验中，分类的整体准确率平均为 93%。而 $N=4$ 时，分类的整体准确率会有所降低，但仍能取得平均 90%的整体准确率。

然后本文分析了在不同 N 取值条件下，机器的处理时间和内存消耗的情况，如图 10~图 13 所示。图 10 所示为在不同 N 取值条件下，流缓存时间 t_{cache} 的累计分布情况。当 $N=6$ 时，有 52%的流缓存时间小于 100ms，99%的流缓存时间小于 1 000ms。而当 $N=4$ 时，则 74%的流缓存时间小于 100ms，99%的流缓存时间小于 500ms。这说明较小的 N 取值，可以大大减少流缓存的时间。

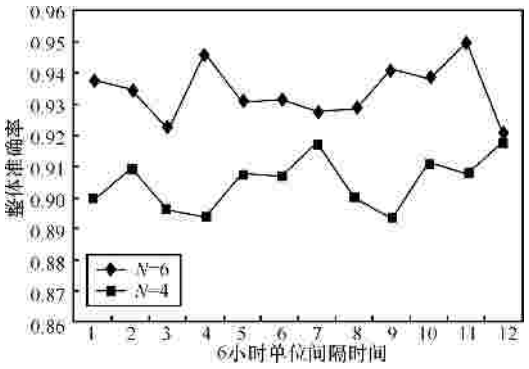

图 9 不同 N 取值下的在线分类整体准确率

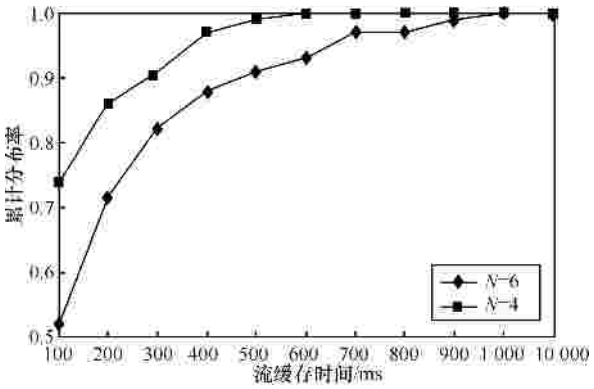

图 10 不同 N 取值下的 t_{cache} 累计分布

图 11 所示为在不同 N 取值条件下，属性计算时间 $t_{\text{attributes}}$ 的累计分布情况。当 $N=6$ 时，有 57%的流属性计算时间小于 100μs，99%的流属性计算时间小于 700μs。而当 $N=4$ 时，则 83%的流属性计算时间小于 100μs，99%的流属性计算时间小于 300μs。较小的 N 取值也可以有效减少流缓存的时间。

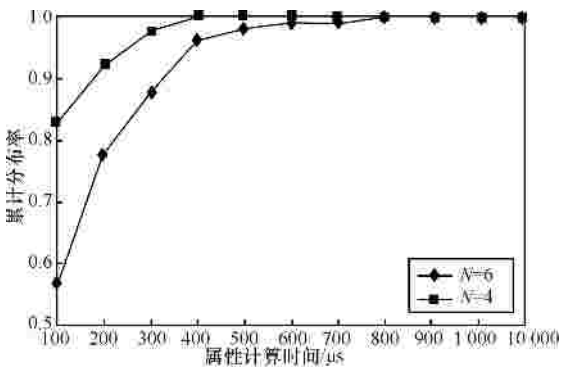

图 11 不同 N 取值下的 $t_{\text{attributes}}$ 累计分布

图 12 所示为在不同 N 取值条件下，流分类时间 $t_{\text{classification}}$ 的累计分布情况。当 N＝6 时，有 83%的流分类时间小于 10μs，99%的流分类时间小于 50μs。而当 N＝4 时，则 89%的流分类时间小于 10μs，99%的流分类时间小于 30μs。较小的 N 取值，虽然可以减少流分类时间，但程度不如对流缓存时间和属性计算时间提高的那样明显。

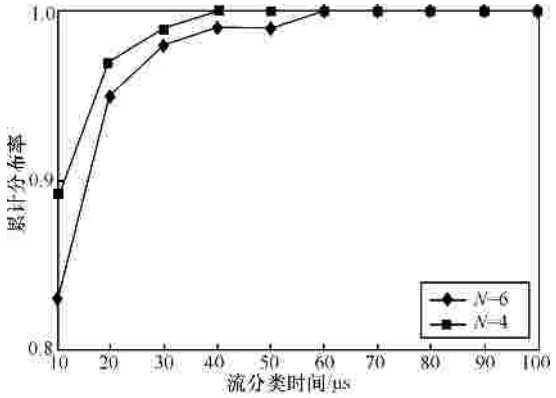

图 12　不同 N 取值下的 $t_{\text{classification}}$ 累计分布

图 13 所示为在不同 N 取值下，系统的内存开销情况。实验每 6 个小时统计一次内存的平均利用率指标，在 N 分别取 6 和 4 的 3 天实验中，一共取得 12 组数据。可以看到 N＝6 的 3 天实验中，实验服务器的内存利用率平均为 71%。而 N＝4 时，内存利用率平均为 32%。这主要是由于较小的 N 取值可以大大降低流缓存的时间，使得每个流得以尽快的处理，释放大量的内存空间。

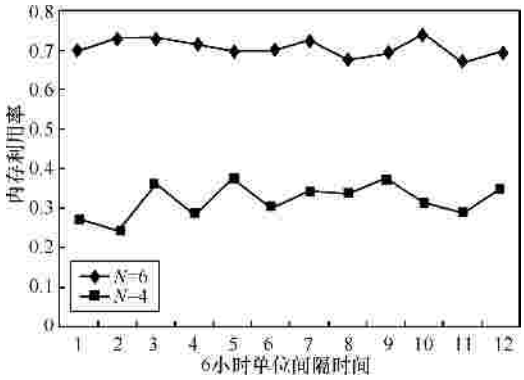

图 13　不同 N 取值下的内存开销

对流量进行在线分类时，实时性是最主要的要求。因此，综合考虑准确性、处理时间和内存开销等因素，N 取 4 是比较理想的结果。此时本文提出的基于最短划分距离的决策树分类方法，可以取得 90%的分类准确性，99%的流处理时间小于 500ms（流缓存时间）+300μs（属性计算时间）+30μs（分类时间），内存利用率平均为 32%。

6　结束语

利用机器学习方法，构造基于流统计特征的网络流量分类模型，是流量分类问题的研究热点。目前大多数的分类模型必须等待流结束后，才能对其进行分类，实时性较差。一些模型从实时性的角度出发，仅仅根据流最初若干分组的方向和大小进行分类，导致其过度依赖分组的到达顺序，稳定性较差。本文除了考虑流最初几个分组的方向、载荷大小特征，主要根据分组的信息熵等特征，采用最短划分距离的方法构建决策树分类模型，对其进行分类。根据在真实网络 Trace 数据集上，以及在线流量分类的实验结果表明，本文方法对 8 种常见的网络应用能够取得较好的分类准确性和综合分类性能。

参考文献：

[1] MOORE A W, PAPAGIANNAKI K. Toward the accurate identification of network applications[A]. Proc of PAM'05[C]. Boston, USA, 2005.41-54.

[2] SEN S, SPATSCHECK O, WANG D. Accurate, scalable in-network identification of P2P traffic using application signatures[A]. Proc of WWW'04[C]. New York, USA, 2004.512-521.

[3] KARAGIANNIS T, BROIDO A, BROWNLEE N, *et al.* Is P2P dying or just hiding[A]. Proc. of GLOBECOM'04[C]. Dallas, USA, 2004.

[4] HAFFNER P, SEN S, SPATSCHECK O, *et al.* ACAS: automated construction of application signatures[A]. Proc of SIGCOMM MineNet'05[C]. New York, USA, 2005. 197-202.

[5] MA J, LEVCHENKO K, KREBICH C, *et al.* Unexpected means of protocol inference[A]. Proc of IMC'06[C]. Rio de Janeiro, Brazil, 2006.313-326.

[6] KARAGIANNIS T, PAPAGIANNAKI K, FALOUTSOS M. BLINC: multilevel traffic classification in the dark[J]. SIGCOMM Computer Communication Review, 2005, 35(4): 229-240.

[7] XU K, ZHANG Z, BHATTACHARYYA S. Profiling Internet backbone traffic: behavior models and applications[A]. Proc of SIGCOMM'05[C]. Philadelphia, USA, 2005.

[8] DAHMOUNI H, VATON S, ROSSE D. A markovian signature-based approach to IP traffic classification[A]. Proc of SIGCOMM MineNet'07[C]. New York, USA, 2007.29-34.

[9] ERMAN J, ARLITT M, MAHANTI A. Traffic classification using clustering algorithms[A]. Proc of SIGCOMM MineNet'06[C]. New York, USA, 2006. 281-286.

[10] MOORE A W, ZUEV D, CROGAN M. Discriminators for use in flow-based classification[R]. RR-05-13, Queen Mary University of London, 2005.

[11] KIM H, CLAFFY K, FOMENKOV M, *et al*. Internet traffic classification demystified: myths, caveats, and the best practices[A]. Proc of ACM CoNEXT'08[C]. New York, USA, 2008.1-12.

[12] NGUYEN T, ARMITAGE G. Training on multiple sub-flows to optimize the use of Machine Learning classifiers in real-world IP networks[A]. Proc of IEEE LCN'06[C]. Tampa, 2006.

[13] BERNAILLE L, TEIXEIRA R., AKODKENOU I. Traffic classification on the fly[J]. ACM SIGCOMM Computer Communication Review, 2006, 36(2): 23-26.

[14] BERNAILLE L, TEIXEIRA R, SALAMATIAN K. Early application identification[A]. Proc of CoNEXT'06[C]. Lisboa, Portugal, 2006. 1-12.

[15] LI J, ZHANG S, LU Y, *et al*. Real-time P2P traffic identification[A]. Proc of GLOBECOM'08[C]. Dallas, USA, 2008. 1-5.

[16] HUANG N, JAI G, CHAO H. Early identification traffic with application characteristics[A]. Proc of ICC'08[C]. Beijing, China, 2008. 5788-5792.

[17] ROUGHAN M, SEN S, SPATSCHECK O, *et al*. Class-of-service mapping for QoS: a statistical signature-based approach to ip traffic classification[A]. Proc of IMC'04[C]. New York, USA, 2004. 135-148.

[18] MOORE A W, ZUEV D. Internet traffic classification using bayesian techniques[A]. Proc of SIGMETRICS'05[C]. Alberta, Canada, 2005. 50-60.

[19] AULD T, MOORE A, GULL S F. Bayesian neural networks for Internet traffic classification[J]. IEEE Transactions on Neural Networks, 2007, 18(1): 223-239.

[20] 徐鹏, 刘琼, 林森. 基于支持向量机的Internet流量分类研究[J]. 计算机研究与发展, 2009, 46(3): 407-414.
XU P, LIU Q, LIN S. Internet traffic classification using support vector machine[J].Journal of Computer Research and Development, 2009, 46(3): 407-414.

[21] WANG R, LIU Y, YANG Y. A new method for P2P traffic identification based on support vector machine[A]. Proc of AIML'06[C]. Sharm El Sheikh, Egypt, 2006. 967-974.

[22] 徐鹏, 林森. 基于 C4.5 决策树的流量分类方法[J]. 软件学报, 2009, 20(10): 2692-2704.
XU P, LIN S. Internet traffic classification using C4.5 decision tree[J]. Journal of Software, 2009, 20(10): 2692-2704.

[23] 王宇, 余顺争. 网络流量的决策树分类[J]. 小型微型计算机系统, 2009, 30(11): 2150-2156.
WANG Y, YU S Z. Internet traffic classification based on decision tree[J]. Journal of Chinese Computer Systems, 2009, 30(11): 2150-2156.

[24] WILLIAMS N, ZANDER S, ARMITAGES G. A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification[J]. SIGCOMM Computer Communication Review, 2006, 36(5): 5-16.

[25] MCGREGOR A, HALL M, LORIER P, *et al*. Flow clustering using machine learning techniques[A]. Proc of PAM'04[C]. Antibes Juanles- Pins, France, 2004. 205-214.

[26] ERMAN J, ARLITT M, MAHANTI A. Traffic classification using clustering algorithms[A]. Proc of SIGCOMM MineNet'06[C]. Pisa, Italy, 2006. 281-286.

[27] ERMAN J., ARLITT M., MAHANTI A. Internet traffic identification using machine learning[A]. Proc of GLOBECOM'06[C]. San Francisco, USA, 2006.

[28] YUAN J, LI Z, YUAN R. Information entropy based clustering method for unsupervised Internet traffic classification[A]. Proc of ICC'08[C]. Beijing, China, 2008. 1588-1592.

[29] ERMAN J, ARLITT M, MAHANTI A. Offline/real-time traffic classification using semi-supervised learning[J]. Performance Evaluation. 2007, 64(9-12): 1194-1213.

[30] QUINLAN R. Discovering Rules from Large Collections of Examples: a Case Study[M]. Expert Edinburgh University Press, 1979.

[31] QUINLAN R. C4.5: Programs for Machine Learning[M]. San Francisco, USA. Morgan Kaufmann Publishers Inc, 1993.

[32] LOPEZ R, MANTARAS D. A Distance-based attribute selection measure for decision tree induction[J]. Machine Learning, 1991, 6(1): 81-92.

[33] LBNL traces[EB/OL].http://ee.lbl.gov/anonymized-traces.html.

[34] CROTTI M, DUSI M, GRINGOLI F, *et al*. Traffic classification through simple statistical fingerprinting[J]. ACM SIGCOMM Computer Communication Review, 2007, 37(1):5-16.

[35] WIDE Project[EB/OL]. http://mawi.wide.ad.jp/mawi/.

[36] Tcptrace official homepage[EB/OL]. http://www.tcptrace.org.

[37] Weka 3: Data mining software in Java[EB/OL]. http://www.cs. waikato. ac.nz/ml/weak.

[38] Analyzer: a public domain protocol analyzer[EB/OL]. http://analyzer. polito.it/.

[39] LI W, CANINI M, MOORE A. Efficient application identification and the temporal and spatial stability of classification schema[J]. Computer Networks, 2009, 53(6): 790-809.

[40] CALLADO A, KELNER J, SADOK D, *et al*. Better network traffic identification through the independent combination of techniques[J]. Journal of Network and Computer Applications, 2010, 33(10): 433-446.

作者简介：

杨哲（1978-），男，江苏苏州人，博士，苏州大学讲师，主要研究方向为网络测量与管理、流量分类与识别。

李领治（1977-），男，山东德州人，博士，苏州大学讲师，主要研究方向为网络体系结构和网络安全。

纪其进（1974-），男，安徽明光人，博士，苏州大学讲师，主要研究方向计算机网络与分布式系统的设计、建模与分析。

朱艳琴（1964-），女，江苏苏州人，博士，苏州大学教授，主要研究方向为计算机网络和信息安全。