基于 C5.0 决策树的 NAT 设备检测方法

石志凯[1] 朱国胜[1,2] 雷龙飞[1] 陈 胜[1] 镇 佳[1] 吴善超[1] 吴梦宇[1]
(湖北大学计算机与信息工程学院 武汉 430062)[1] (湖北省教育信息化工程技术研究中心 武汉 430062)[2]

摘 要 网络地址转换 NAT 对外网隐藏了内网的结构,这一方面给非法终端隐匿接入提供了便利,对网络造成潜在威胁;另一方面,用户也可通过 NAT 私自共享网络,直接损害网络运营者的利益。有效检测 NAT 设备,对网络安全管控、运营管理具有重要作用。文中对现有 NAT 检测技术进行了分析与比较,阐述了各自的优缺点和适用条件;提出了一种利用上层应用的特征和训练数据构建 C5.0 决策树的 NAT 设备检测方法。真实网络环境下的实验表明,所提方法能有效识别 NAT 设备。

关键词 NAT,NAT 检测,C5.0 决策树
中图法分类号 TP181 **文献标识码** A

NAT Device Detection Method Based on C5.0 Decision Tree

SHI Zhi-kai[1] ZHU Guo-sheng[1,2] LEI Long-fei[1] CHEN Sheng[1] ZHEN Jia[1] WU Shan-chao[1] WU Meng-yu[1]
(School of Computer and Information Engineering, Hubei University, Wuhan 430062, China)[1]
(Hubei Province Engineering Technology Research Center for Education Informationization, Wuhan 430062, China)[2]

Abstract NAT hides the internal network structure to the external network. On the one hand, it offers access to the illicit terminal facilitates, causing potential threats to the network. On the other hand, users can also privately share networks through NAT, which directly harm the interests of network operators. Effective detecting NAT devices plays an important role in network security and controlling, network operation and management. This article analyzed and compared the current NAT detection technologies. The advantages, disadvantages and the applicable conditions of each technologies were described. A C5.0 decision tree based NAT device detection method using features of the upper-level applications and training data was proposed in this paper. The experiments with real network traffic data show that the model can identify NAT device effectively.

Keywords NAT, NAT detection, C5.0 decision tree

1 引言

随着互联网的不断发展,接入互联网的设备和互联网用户急速增加,据统计,目前全球网民有 36 亿,接入互联网的设备有 203.5 亿[1-2]。大量设备的接入造成有限 IPv4 地址急剧短缺,网络地址转换(Network Address Translation, NAT)技术发挥着不可替代作用[3]。NAT 技术在有效缓解 IPv4 地址短缺及核心路由表规模的同时,也存在很多弊端,用户可以很便利地利用 NAT 进行再次组网而不影响对互联网的访问,增加了网络使用的灵活性。NAT 设备的使用会造成数据包延时增加并且无法实现端对端的跟踪,而更重要的是,网络服务提供商(Internet Service Providers, ISP)无法知道内网的规模和结构,用户可能私自提供网络服务,直接损害网络经营者的利益;同时,非法 NAT 设备的接入,可能会造成网络资源的不规范使用,甚至给攻击者提供了隐匿主机的便利,给网络管理造成困难甚至带来潜在危险。

NAT 设备的检测对网络服务提供商和网络环境的安全维护有着重要作用。目前,NAT 设备的检测主要分为两大类:一类是基于协议标识的检测技术,通过各种协议的标识信息,能快速识别出是否为 NAT 设备,但使用的字段可能遭到修改,对于加密协议,无法获取所需的字段;另一类是基于流量特征的检测技术,依据 NAT 设备下多个用户使用的流量特征与单个用户使用的流量特征的差异,通常利用机器学习算法建立二元分类模型,对于 NAT 下有多个活跃用户的情况,其能有效识别出 NAT 设备,但模型较依赖网络环境的实际情况,网络带宽和用户实际使用流量的变化会直接影响模型效率,因此模型需要根据网络环境的变化不断更新训练数据。

本文提出一种基于上层应用特征的 NAT 设备检测方法,通过上层应用特征建立 C5.0 决策树分类模型进行 NAT 设备检测。选取的主要特征有用户浏览的网站和用户使用过的应用程序。网站和应用程序的识别是根据预先建立的应用

本文受赛尔网络下一代互联网技术创新项目(NGII20150101)资助。
石志凯(1992-),男,硕士生,主要研究方向为网络流量分析;朱国胜(1972-),男,博士,教授,主要研究方向为下一代互联网、软件定义网络,E-mail:zhuguosheng@hubu.edu.cn(通信作者);雷龙飞(1993-),男,硕士生,主要研究方向为网络流量分析;陈 胜(1994-),男,硕士生,主要研究方向为大数据分析;镇 佳(1992-),女,硕士生,主要研究方向为文本分析;吴善超(1994-),男,硕士生,主要研究方向为数据挖掘;吴梦宇(1994-),硕士生,主要研究方向为网络大数据分析、数据挖掘。

识别库进行匹配，应用识别库搜集了常用网站和常用 app，因此在实际网络环境中，即使使用了 HTTPS 等加密协议，也能匹配出其他访问信息，建立模型训练特征向量。

本文第 2 节介绍 NAT 检测技术的现状，对各种检测技术进行了分析和比较；第 3 节给出了本文提出的基于应用的 NAT 检测方法，包括模型与算法步骤；第 4 节为实验验证，即利用实际数据对检测方法进行检验与评估。

2 NAT 设备检测技术的现状

2.1 基于协议标识的检测技术

基于协议标识的检测技术，即根据某些协议的某些字段信息进行识别的技术，其主要包括两方面，一是某些字段不会因为 NAT 的使用而发生改变，另一方面是某些字段因为 NAT 设备的使用而表现出特有的变化特征。

Bellovin[4] 提出一种利用 IP 数据包头部的 IPid 字段识别 NAT 设备以及估计 NAT 下有多少设备接入的方法。IPid 用于重组被分割的数据包，在同一个源和目的地址之间，每个数据包中该字段的值是独一无二的。Bellovin 等利用一种算法能对 IPid 的值分组，形成多个 IPid 的序列，每个序列即代表一个真实主机。但算法的不足之处在于 IPid 值可能会产生冲突，并且字段的值也不一定是有序自增，甚至有些 NAT 设备会将该值重设为零。这些都直接影响着算法的准确性。为解决 IPid 方法的缺陷，Straka 等[5] 提出结合 IPid 和 TTL 字段的方法。大部分操作系统拥有自己默认的 *TTL* 值，并且每经过一个 NAT 设备，*TTL* 值将减 1，由此可以轻易检测其是否是 NAT 设备。但 *TTL* 值仍存在被主机修改的可能。在此基础上，Maier 等[6] 利用 HTTP 协议的 user-agent 字符串识别出不同的操作系统类型，根据操作系统的不同，能预先知道 *TTL* 的默认值，结合识别 TTL 序列的算法，能提高识别的准确率。但 user-agent 字段同样可被主机修改，同时对于 HTTPS 协议，无法获取 user-agent 字段的值。

Mongkolluksamee 等[7] 提出利用 TCP/IP 协议字段的方法，利用 IPid 序列、TCP 序列和源端口号来识别 NAT 设备。这些字段都有自增的特点，利用自增的规律能得出多个不同的序列，根据序列的多少来判断是否为 NAT 设备。该方法的主要缺点是对 OpengBSD 系统的识别会将多个主机计算为一个主机。Zhang 等[8] 提出一种混合多个字段的方法，采用 cookieID，同时应用 ID 和 user-agent 字符串，但该方法依然存在同样的缺点。

2.2 基于流量特征的检测技术

一般一个 NAT 设备下有多个用户，即一个 IP 地址对应多个用户，而非 NAT 情况下，一个 IP 即对应一个用户。依据多个用户使用的流量特征和单个用户使用的流量特征的不同，利用机器学习算法可有效识别 NAT 设备。

Rui 等[9] 提出了基于流量特征的 SVM 检测方法，采用了 UDP 包个数和 TCP 包个数等 8 个特征。Dietz 等[10] 同样利用 SVM 和 C4.5 决策树等机器学习算法检测 NAT 设备。与此类似，Gokcen 等[11] 利用朴素贝叶斯和 C4.5 决策树，根据 7 个流量特征进行 NAT 设备的识别。除了利用流量特征，Komarek 等[12] 提出利用 HTTP 协议的相关特征，利用 SVM 进行 NAT 设备的识别。

总之，基于协议标识的检测技术大都从数据包字段信息获取标识，一般从 IP，TCP 或 HTTP 协议中的某些字段获得，如 TTL、IPid、UA 字符串和端口号等。通过这些字段的信息，标识出一个 IP 地址下的多个相似的包序列，从而可以估计 IP 地址对应的是否是 NAT 设备以及 NAT 下设备的个数。识别出操作系统的类型和浏览器版本等信息，判断有多少种不同的设备在一个 IP 下工作，从而判断是否为 NAT 设备。通常为了获得更高的识别率，会结合多个字段进行识别和判断。

基于某个标识字段的识别方法存在的问题主要有：IPid 可能被设置成一些不可预知的数或是表现出一种随机的变化；*TTL* 值也可能被设置为不是操作系统的默认值。有些主机为避免被检测到，会刻意修改数据包头部这些字段的值，从而影响检测效果。同时，对于加密了头部的数据包，如 HTTPS 协议，某些协议字段无法获取。尽管如此，在实际网络环境中，直接利用流量标识的一些信息字段进行 NAT 设备的检测，仍然有较高的识别率。

基于流量行为特征的检测技术，一般从流量特征来获取标识信息。依据 NAT 设备的流量特征和单个用户产生的网络流量特征的差异性识别出 NAT 设备。通常情况下，NAT 设备的流量，即多个设备或者用户的流量，会明显高于一个用户使用的流量。利用的流量特征信息包括流量的大小、数据包的个数、TCP 包的个数、UDP 包的个数和 DNS 请求包的个数等。一般结合机器学习算法，利用训练数据进行训练得到识别模型。如果一个 NAT 设备下有超过 5 个活跃设备，利用这类识别方法识别出 NAT 设备的准确率接近 100%。但对于 NAT 设备下少于 4 个设备的情况，准确度会明显下降。同时，为适应不同的网络环境，因带宽的差异或用户使用网络的习惯差异，需适时训练更新模型，以适应不同的网络环境。

本文基于网络用户使用的应用的特征分析，提出基于应用特征的 NAT 设备识别方法。该方法不完全依赖某个数据包的某个特征，也不需要根据网络环境的变化而频繁学习。

表 1 对比了各种方法。

表 1 各方法对比说明

方法	适用加密协议	适用多种操作系统	可伪造	多层 NAT	需训练
协议标识	×	×	√	×	×
流量特征	√	√	×	√	√
应用特征	√	√	×	√	√

3 基于应用的 NAT 识别技术

3.1 直接基于应用的识别

基于应用的 NAT 识别是利用上层应用的相关信息进行 NAT 设备的识别技术。利用上层应用的相关信息进行识别有以下 3 个优点：

1）NAT 设备很难通过修改应用层的相关信息来躲避检测；

2）NAT 下的用户也很难获得和修改应用的源码等相关信息来躲避检测；

3）应用的开发者没有为躲避 NAT 的检测去修改应用程序的动机。

Bi 等[13] 提出一种利用即时通讯（Instant Messaging，IM）应用来检测 NAT 设备的方法，该方法利用即时通讯软件的普及性，以 MSN 和 Google talk 为例，检测即时通信应用所用

信道使用的数量，以此判断检测对象是否为 NAT 设备。利用即时通讯软件有几个优点：1）使用的人数较多；2）使用的端口通常为注册端口。但这种单纯利用某种或某些应用的方法有很大局限性。就目前国内流行的即时通讯软件 QQ 而言，有很多终端如 Linux 系统终端，就很可能不会有 QQ 用户登录，而在同一个 Windows 终端上，又可同时登录多个 QQ 账号。此外，这种检测方式不能完全排除用户不用或者刻意不用某种应用的情况。

鉴于直接运用某种或某类应用进行识别的方法存在的缺陷，本文采用的方法结合了多个应用，并提取应用的多个维度的特征，利用这些特征，通过训练数据训练得到基于 C5.0 决策树的 NAT 设备识别模型。

3.2　应用特征的选取

本文选用了用户浏览的网页和使用的应用程序的相关信息，包括使用的 app 数量、app 所属类的数量、app 使用时长、Web 浏览数量、Web 浏览所属类数量和 Web 浏览时长 6 个字段。因为用户访问的网站和 app 种类过多，所以识别网站的 host 和 app 是由预先建立好的应用识别库确定，一方面用户访问的网站和 app 无法全部统计，另一方面应用识别库搜集了大部分常用 app 和 Web 的信息，能有效标识用户的访问信息，提高数据处理的效率。

Web 访问的信息库包含近 3000 个常用站点，部分样例如表 2 所列。

表 2　Web 标识库样例

站点名	host	站点类
中关村在线	zol.com.cn	IT 网站
太平洋电脑网	pconline.com.cn	IT 网站
脚本之家	jb51.net	IT 网站
CSDN	csdn.net	IT 网站
站长之家	chinaz.com	IT 网站
天极-Yesky	yesky.com	IT 网站
博客园	cnblogs.com	IT 网站

app 访问信息库的部分样例如表 3 所列。

表 3　app 标识库样例

app	app 类
MomoChat	社区交友
live	网络直播
LOFTER	社区交友
Letv	在线视频
Youku	在线视频
kugou	在线音乐
kwai	在线视频
hearthstone	游戏服务
JD	网络购物
YYMobile	网络直播

在流量采集阶段对 Web 和 app 进行识别，app 通过端口和特定协议进行识别，Web 通过 HTTP 协议中的 hosts 字段进行匹配；同时，统计出各个 IP 下所识别出的 app、网站名，以及访问时长。

根据提取的数据计算出所需特征，如表 4 所列。根据湖北大学校园实际网络环境部署的情况，有分配给学生公寓使用的 NAT 设备地址，也有给公共环境接入无线网络的独立用户地址，据此可以为数据做出是否为 NAT 设备的标记，得到训练数据，训练数据的格式如表 5 所列。

表 4　数据集属性与说明

属性	描述
IP	待识别 IP 地址
A_n	app 数量
A_{cn}	app 所属类数量
A_t	app 使用总时长
W_n	Web 浏览数量
W_{cn}	Web 浏览所属类数量
W_t	Web 浏览总时长

表 5　用于训练的数据属性

属性	A_n	A_{cn}	A_t	W_n	W_{cn}	W_t	class
类型	Num	Num	Num	Num	Num	Num	Bool

3.3　基于应用的 C5.0 决策树 NAT 识别模型

根据所选取的应用特征和标记生成 C5.0 决策树的训练集和测试集，训练集用于 C5.0 决策树模型的学习，利用测试集检验识别的准确率。模型识别的流程如图 1 所示。

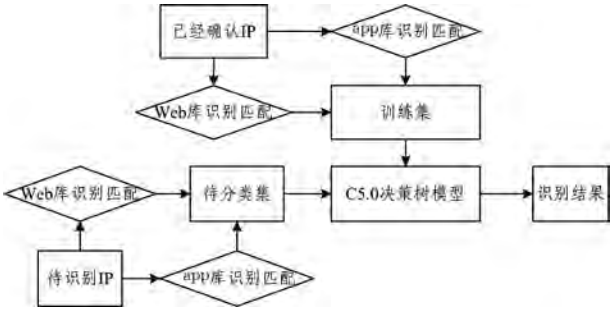

图 1　C5.0 决策树识别模型

设样本集 D 按属性 A 的 n 个不同属性值划分为 D_1，D_2，…，D_n 共 n 个子集，则分割信息量表示为：

$$split_{info_A}(D)=-\sum_{j=1}^{n}\frac{|D_j|}{|D|}\log(\frac{|D_j|}{|D|}) \tag{1}$$

信息增益率表示为：

$$gain\_ratio(A)=\frac{gain(A)}{split\_info(A)} \tag{2}$$

利用训练集生成分类模型是检测方法的核心步骤，本文采用了 10 个数据集，分别获取训练集，记为 $S=\{D_1,D_2,\cdots,D_n\}$，每个训练集除包括 IP 和对应类别外，还包括使用的 app 数量、app 所属类的数量、app 的使用时长、Web 浏览数量、Web 浏览所属类数量和 Web 浏览时长 6 个字段，分别用 A_n，A_{cn}，A_t，W_n，W_{cn}，W_t 表示。对于某个训练集 S_s，将各个字段分别划分为多个取值区间，记为 C_1，C_2，…，C_m 共 m 个子集，记 $P(C_q)=|C_p|/|S_s|(1\leqslant p\leqslant m)$。则该训练集对分类的平均信息量为：

$$H(S_s)=-\sum_{p=1}^{m}P(C_p)\log_2 P(C_p) \tag{3}$$

对于其中的任意字段，如 A_n，假设有 t 个不同取值 a_q $(1\leqslant q\leqslant t)$，那么根据 A_n 的不同取值，可以将 S_s 划分为 S_1，S_2，…，S_t 共 t 个子集，同时可以将 C_1，C_2，…，C_m 划分为 $m\times t$ 个子集，每个子集 C_{pq} 表示在 $A_n=a_q$ 的条件下属于第 p 类的样本集合，其中，$1\leqslant p\leqslant k(1\leqslant q\leqslant t)$。对 A_n 进行划分后，样本集对分类的平均信息量为：

$$H(S_s/A_n)=-\sum_{q=1}^{t}P(C_q)\left[-\sum_{p=1}^{m}P(C_{pq})\log_2 P(C_{pq})\right] \tag{4}$$

其中，$P(C_q)=\sum_{p=1}^{m}|C_{pq}|/|S_s|$，$P(C_{pq})=|C_{pq}|/|S_s|$，可计算出 A_n 对 S_s 进行划分的信息增益 $f_G(S_s,A_n)$ 为：

$$f_G(S_s,A_n)=H(S_s)-H(S_s/A_n) \tag{5}$$

其表示划分后不确定性的下降程度。使用属性 A_i 划分 S 的信息增益率为信息增益量与分割信息量的比值，因此可以得到：

$$f_{GR}(S_s,A_n)=\frac{f_G(S_s,A_n)}{f_{sp}(S_s,A_n)} \tag{6}$$

其中，f_{sp} 为分割信息量。根据计算得到的信息增益率，算法由大到小进行选择，从决策树顶端开始，自上而下建立，待完整树生成以后，C5.0 采用后剪枝法从叶子节点逐层向上修剪。修剪依据为各子节点加权误差是否大于父节点的加权误差，如果大于则剪出。如此便形成了最终的决策树模型。

4 实验分析

4.1 数据说明与处理

本文实验采用的网络流量来源于湖北大学校园网，利用端口镜像获取 4 个千兆端口的上下行流量，采集时间为 2017 年 7 月 10 日到 2017 年 7 月 19 日。

采集工具使用的是美国伯克利大学开发的开源工具 bro[14]，bro 是一个强大的网络分析和网络监控框架，通过 bro 可以获取多种协议的流量信息，包括 HTTP，TCP，UDP 等常用协议，其以日志文件的形式存储流量信息。

数据集信息说明如表 6 所列。

表 6 数据集信息

数据集	IP 数	抓取时间
DataSet1	197	2017-07-10
DataSet2	204	2017-07-11
DataSet3	214	2017-07-12
DataSet4	197	2017-07-13
DataSet5	200	2017-07-14
DataSet6	190	2017-07-15
DataSet7	207	2017-07-16
DataSet8	187	2017-07-17
DataSet9	218	2017-07-18
DataSet10	173	2017-07-19

4.2 实验平台

本文流量采集使用的是 3 台服务器搭建的 bro 分布式采集环境，数据挖掘工具采用 Weka-3.6.13[15]。Weka 是 JAVA 环境下开发的开源机器学习以及数据挖掘工具。运行环境是普通 PC，内存为 8GB，处理器为 coreI7-6500U，操作系统为 Windows。

4.3 实验与结果分析

针对每个数据集，取 60% 的数据作为训练集，其余为测试集，分别训练模型并测试各个数据集的准确率，结果如表 7 所列。

表 7 模型识别率

数据集	IP 数	模型识别率/%
DataSet1	197	97.4684
DataSet2	204	85.3659
DataSet3	214	94.1860
DataSet4	197	89.8734
DataSet5	200	87.5000
DataSet6	190	93.4211
DataSet7	207	95.1807
DataSet8	187	98.6667
DataSet9	218	98.8506
DataSet10	173	100.0000

为验证模型的稳定性，即模型适应不用环境的能力，以数据集 DataSet1 作为训练集，其他数据集作为测试集，结果如表 8 所列。

表 8 以 DataSet1 为训练集的模型识别率

数据集	IP 数	模型识别率/%
DataSet2	204	84.0000
DataSet3	214	92.0000
DataSet4	197	81.0000
DataSet5	200	82.0000
DataSet6	190	91.0000
DataSet7	207	87.0000
DataSet8	187	89.0000
DataSet9	218	93.0000
DataSet10	173	92.0000

由此可知，即使对于时间间隔较长的网络环境，模型训练一次后仍具有较好的识别率，并保持比较好的稳定性。

5 未来研究

本文利用上层应用的特征，结合 C5.0 决策树算法，实现了 NAT 设备检测方法，克服了直接利用某些协议的某些字段直接进行识别容易被修改的缺陷。其相比直接运用流量的特征进行机器学习的分类更加具有稳定性，能在实际网络环境中准确识别 NAT 设备。但算法所用特征字段需要与预先建立的标识库进行匹配，算法的准确性和效率也依赖数据库的数据。对预先建立的数据进行拓展和优化，实现数据的自动化处理和算法的自动匹配识别，是下一步研究的重点。

参 考 文 献

[1] STATISTA. Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions)[OL]. [2017-05-14]. https://www.statista.com/statistics/471264/iot-numberof-connected-devices-worldwide.

[2] Internet Live Stats. Internet Users[OL]. [2017-0514]. http://www.internetlivestats.com/ internet-users/#trend.

[3] NESSETT D M,GRABELSKY D,BORELLA M S,et al. Method and system for locating network services with distributed network address translation:U.S. Patent 6,055,236[P]. 2000-4-25.

[4] BELLOVIN S M. A technique for counting NATted hosts[C]// Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment. ACM,2002:267-272.

[5] STRAKA K,MANES G. Passive detection of nat routers and client counting[M]// Advances in Digital Forensics II. Springer, Boston,MA,2006:239-246.

[6] MAIER G,SCHNEIDER F,FELDMANN A. NAT usage in residential broadband networks[C]// International Conference on Passive and Active Network Measurement. Springer,Berlin, Heidelberg,2011:32-41.

[7] MONGKOLLUKSAMEE S,FUKUDA K,PONGPAIBOOL P. Counting NATted hosts by observing TCP/IP field behaviors [C]// 2012 IEEE International Conference on Communications (ICC). IEEE,2012:1265-1270.

[8] ZHANG B,GUAN Y,NIU W,et al. A hybrid packet clustering approach for NAT host analysis[C]// 2015 IEEE International Conference on Communication Software and Networks(ICCSN).

IEEE，2015：432-438.

[9] RUI L，HONGLIANG Z，YANG X，et al. Remote NAT detect algorithm based on support vector machine[C]// 2009 International Conference on Information Engineering and Computer Science. 2009.

[10] ABT S，DLETZ C，BAIER H，et al. Passive remote source nat detection using behavior statistics derived from netflow[C]// IFIP International Conference on Autonomous Infrastructure，Management and Security. Springer，Berlin，Heidelberg，2013：148-159.

[11] GOKCEN Y，FOROUSHANI V A，HEYWOOD A N Z. Can we identify NAT behavior by analyzing Traffic Flows? [C]// Security and Privacy Workshops (SPW)，2014 IEEE. IEEE，2014：132-139.

[12] KOMÁREK T，GRILL M，PEVNY T. Passive NAT detection using HTTP access logs[C]// 2016 IEEE International Workshop on Information Forensics and Security (WIFS). IEEE，2016：1-6.

[13] BI J，ZHAO L，ZHANG M. Application presence fingerprinting for NAT-aware router[C]// Knowledge-Based Intelligent Information and Engineering Systems. Springer Berlin/Heidelberg，2006：678-685.

[14] The Bro Network Security Monitor[OL]. http://www.bro.org.

[15] HOLMES G，DONKIN A，WITTEN I H. Weka：A machine learning workbench[C]// Proceedings of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems. IEEE，1994：357-361.

（上接第 309 页）

3 个阶段：截止时间重分配、任务调度选择排序和基于电压/频率等级的资源选择。通过这种三阶段方式，算法最终可以得到满足工作流 QoS 约束且带有合适电压/频率等级的最优工作流调度方案，使得工作流的总体能耗达到最小。通过随机工作流结构和一种现实科学工作流结构的仿真实验证明了算法的有效性和可行性。进一步的研究将考虑更多维度的 QoS 约束，如预算约束、安全约束等，以研究多 QoS 约束下的能效调度问题。

参 考 文 献

[1] BUYYA R，YEO S，VENUGOPAL S，et al. Cloud computing and emerging IT platforms：Vision，hype，and reality for delivering computing as the 5th utility[J]. Future Generation Computer Systems，2011，25(6)：599-616.

[2] LIU L，ZHANG M，LIN Y，et al. A survey on workflow management and scheduling in cloud computing[C]// 14th IEEE/ACM International Symposium on Cluster，Cloud and Grid Computing. USA：IEEE Press，2014：837-846.

[3] ARROBA P，RISCO-MART J L，ZAPATER M，et al. Server power modeling for run-time energy optimization of cloud computing facilities[C]// International Conference on Sustainabiility and building. 2014：401-410.

[4] TOPCUOGLU H，HARIRI S，WU M Y. Performance-effective and low-complexity task scheduling for heterogeneous computing[J]. IEEE Transactions Parallel Distributed Systems，2012，13(3)：260-274.

[5] SELVI S，MANIMEGALAI D. Task Scheduling Using Two-Phase Variable Neighborhood Search Algorithm on Heterogeneous Computing and Grid Environments[J]. Arabian Journal for Science and Engineering，2015，40(3)：817-844.

[6] LEE Y C，HAN H，ZOMAYA A Y，et al. Resource-efficient workflow scheduling in clouds[J]. Knowledge-Based Systems，2015，80(C)：153-162.

[7] ABUDHAGIR U S，SHANMUGAVEL S. A Novel Dynamic Reliability Optimized Resource Scheduling Algorithm for Grid Computing System[J]. Arabian Journal for Science and Engineering，2014，39(10)：7087-7096.

[8] ARABNEJAD H，BARBOSA J G. A Budget Constrained Scheduling Algorithm for Workflow Applications[J]. Journal of Grid Computing，2014，12(4)：665-679.

[9] LEE Y C，ZOMAYA A Y. Minimizing Energy Consumption for Precedence-Constrained Applications Using Dynamic Voltage Scaling[C]// IEEE/ACM International Symposium on CLUSTER Computing and the Grid. IEEE Computer Society，2009：92-99.

[10] ZONG Z，MANZANARES A，RUAN X，et al. EAD and PEBD：Two Energy-Aware Duplication Scheduling Algorithms for Parallel Tasks on Homogeneous Clusters[J]. IEEE Transactions on Computers，2011，60(3)：360-374.

[11] BASKIYAR S，ABDEL-KADER R. Energy aware DAG scheduling on heterogeneous systems[J]. Cluster Computing，2013，13(4)：373-383.

[12] 景维鹏，吴智博，刘宏伟，等. 多 DAG 工作流在云计算环境下的可靠性调度方法[J]. 西安电子科技大学学报(自然科学版)，2016，43(2)：83-88.

[13] 王润平，陈旺虎，段菊. 一种科学工作流的云数据布局与任务调度策略[J]. 计算机仿真，2015，32(3)：421-426.

[14] 曹斌，王小统，熊丽荣，等. 时间约束云工作流调度的粒子群搜索方法[J]. 计算机集成制造系统，2016，22(2)：372-380.

[15] 杨玉丽，彭新光，黄名选，等. 基于离散粒子群优化的云工作流调度[J]. 计算机应用研究，2014，31(12)：3677-3681.

[16] CHEN W，DEELMAN E. WorkflowSim：a toolkit for simulating scientific workflows in distributed environments[C]// IEEE 8th International Conference on E-Science (e-Science). IEEE，2012：1-8.

[17] JUVE G，CHERVENAK A，DEELMAN E，et al. Characterizing and profiling scientific workflows[J]. Future Generation Computer Systems，2013，29(3)：682-692.