一种基于 Cloud-P2P 计算模型的恶意代码联合防御网络*

徐小龙[1a,2]，吴家兴[1a]，杨 庚[1b]

(1. 南京邮电大学 a. 计算机学院，b. 计算机技术研究所 南京 210003；2. 中国科学院软件研究所 信息安全国家重点实验室，北京 100190)

摘 要：针对目前的反病毒系统在应对恶意代码时通常具有的滞后性，提出并构建了一种新颖的基于 Cloud-P2P 计算模型的恶意代码联合防御网络。Cloud-P2P 计算模型将云计算与对等计算进行有机融合。恶意代码联合防御网络系统中的集群服务器与用户终端群体联合组成了一个高安全防御网，协同防御恶意代码，并快速产生群体免疫力。为了提高系统的性能表现，提出适用于 Cloud-P2P 融合计算环境的两种基于分布式哈希表的层次式网络结构 C-DHT 和 D-DHT，并通过引入移动 agent 技术实现了恶意代码联合防御网络中的疫苗 agent 和巡警 agent。基于 Cloud-P2P 计算模型的恶意代码联合防御网络具有负载均衡、反应快捷、防御全面和兼容性良好等性能表现。

关键词：云计算；对等计算；恶意代码；反病毒；云安全

中图分类号：TP393 **文献标志码**：A **文章编号**：1001-3695(2012)06-2214-04

doi:10.3969/j.issn.1001-3695.2012.06.056

Malicious code united-defense network based on Cloud-P2P model

XU Xiao-long[1a,2], WU Jia-xing[1a], YANG Geng[1b]

(1. *a. College of Computer, b. Institute of Computer Technology, Nanjing University of Posts & Telecommunications, Nanjing* 210003, *China*; 2. *State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing* 100190, *China*)

Abstract: The current anti-virus systems are usually unable to respond to endless emerging malicious codes in time. To solve this problem, this paper proposed and constructed a new malicious code united-defense network based on the Cloud-P2P computing model. The Cloud-P2P model integrated the cloud computing and the P2P computing systems together organically. Servers and user terminals in the malicious code united-defense network carry out their own duties, forming a high-security collaborative defense network against malicious codes and obtaining the whole group immunity quickly. It also proposed two kinds of hierarchical network topology, C-DHT and D-DHT, which were based on the distributed hash table technology and suitable for the Cloud-P2P computing model. By the introduction of mobile agent technology, realized vaccine agent and patrol agent of the malicious code defense united-network. The malicious code united-defense network based on the Cloud-P2P computing model has ideal performances, such as the network load balance, the rapid response, the comprehensive defense and the good compatibility.

Key words: cloud computing; P2P computing; malicious code; anti-virus; cloud security

目前对于各种新型的恶意代码的反应与处理，一般都要等到恶意代码对网络及主机进行了大规模的破坏后才能被发现，然后再给出相应的解决方案，这种机制显然具有较大的滞后性[1]。目前的“云安全”系统[2]过分强调服务器构成的“云”对恶意代码的防御，而忽视用户终端设备的能力，容易导致性能瓶颈问题；也不提供用户与用户之间的相互通报恶意代码信息的渠道，使得恶意代码的网络生命周期仍然较长，造成的危害和损失仍然较大。另外许多反病毒软件扫描发现了恶意代码却无力清除，甚至反被恶意代码中止，随即系统和数据被破坏[3]，这也是难以解决的问题。

1 Cloud-P2P 计算模型

目前的云计算(cloud computing)[4~6]侧重于利用网络服务器端的资源，而对等计算(P2P computing)[7]则侧重于利用网络边缘节点的资源。虽然云计算与 P2P 计算有所不同，但从本质上来说都是要解决分布式计算或资源共享问题。本文提出的 Cloud-P2P 计算模型是将云计算和对等计算这两种分布式计算技术有效地融合、互连互通，应用于网络计算系统，可

收稿日期：2011-10-09；**修回日期**：2011-11-16 **基金项目**：国家自然科学基金资助项目(60873231)；国家教育部高等学校博士学科点专项科研基金资助课题(20093223120001)；中国博士后科学基金资助项目(2011M500095)；江苏省自然科学基金资助项目(BK2011754，BK2009426)；江苏省科技支撑计划资助项目(BE2009158)；江苏省高校自然科学研究项目(09KJB520010)；信息安全国家重点实验室开放课题(03-01-1)；江苏高校优势学科建设工程资助项目(yx002001)；江苏省博士后科研资助计划项目(1102103C)

作者简介：徐小龙(1977-)，男，江苏盐城人，副教授，博士(后)，主要研究方向为计算机软件、分布式计算、信息安全、agent 技术等(xuxl@njupt.edu.cn)；吴家兴(1988-)，男，江苏连云港人，硕士研究生，主要研究方向为计算机软件、分布式计算、可信计算技术；杨庚(1961-)，男，江苏盐城人，教授，博导，博士，主要研究方向为计算机应用、网络计算技术、信息安全技术等.

实现最大范围的协作、通信与资源分享,更好地整合互联网和不同设备上的信息及应用,发挥各种特长,达到应用高效率、低成本的目标。

Cloud-P2P 计算模型对云计算和对等计算的融合体现在:

a)节点的融合。即将云计算中的集群服务器节点与对等计算系统中的用户终端节点统一置于一个分区域的资源池中。模型是由核心管理层、云功能层和 P2P 网络层构成。核心管理层由主服务器节点构成,负责对 Cloud-P2P 计算计算环境中的所有资源进行统筹管理;云功能层由承担面向特定应用的集群服务器节点构成;P2P 网络层由终端节点构成。Cloud-P2P 计算模型基本结构如图 1 所示。

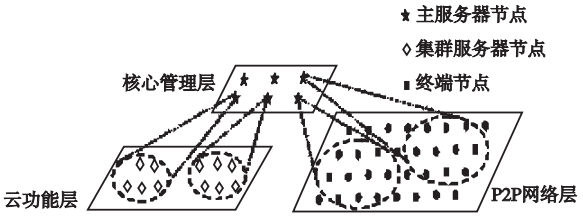

图1　Cloud-P2P计算模型基本结构

b)网络拓扑的融合。云计算系统一般采用中心化拓扑(centralized topology),强调良好的可管理性。Google 云计算系统[8,9]采用的 Server farm 形式就是基于中心化拓扑,使用一个或几个主控服务器(master)加大量集群服务器构成,主控服务器负责管理资源,并监控各节点的状态,进行失效检测和实现负载均衡等。纯粹的 P2P 网络计算系统则强调了去中心化的原则,对于资源定位、路由以及系统的动态性和随机性响应主要依靠节点之间的彼此协作来完成,最有代表性的是基于 DHT(distributed hash table,分布式散列表)技术[10,11]的全分布式结构化拓扑结构。这两种拓扑结构各有优势和不足,例如中心化拓扑容易带来性能瓶颈,单纯的 DHT 的维护机制较为复杂等。在复杂的 Cloud-P2P 计算环境下,显然需要将这两种拓扑结果进行有机地融合,实现更有效的网络拓扑。

除此之外,还有网络协议、通信机制、安全机制和管理机制等方面的融合,限于篇幅,本文不再赘述。

2　恶意代码联合防御网络

2.1　体系架构

本文利用 Cloud-P2P 计算架构来解决网络中的恶意代码问题是保障网络时代信息安全的一个崭新思路。基于 Cloud-P2P 计算模型的恶意代码联合防御网络来获取病毒样本和警报信息、识别和解决新恶意代码彻底改变了节点各自为政的情况,充分发挥 Cloud-P2P 计算环境中服务器集群与终端群体各自的优势,以协同方式实时进行病毒采集、分析、处理和网络系统防御能力的快速整体升级。

基于 Cloud-P2P 计算模型的恶意代码联合防御网络中,整个网络形成了一个恶意代码联合防御整体,服务器与用户终端各自承担的任务包括:

a)服务器作为反病毒监控与解决中心,负责恶意代码的样本与报告收集、恶意代码解决方案(病毒库升级包和系统补丁)的分发、网络节点状态异常监控、多种反病毒引擎(或系统)的集成与协调等。

b)用户终端作为反病毒末梢节点,负责提交恶意代码样本和主机异常状态报告、本地主机系统病毒防御与升级、节点间报警、终端节点间恶意代码解决方案传递等。

服务器集中置于云计算数据中心。不同的反病毒引擎分别侧重于木马、病毒、蠕虫、间谍件、游戏盗号或密码窃取等各种不同的恶意程序。把服务器按照其上安装的引擎将服务器集群分为多个防御区域,另外选择一个服务器节点作为用户终端接入集群服务器的门户节点。用户终端上也可以自主安装相应的反病毒软件,因此可按照所安装的反病毒软件的不同将用户终端分为多个虚拟防御组(virtual defense group,VDG)。这样分组的优点主要在于,可以将基于某特定反病毒软件的恶意代码解决方案在安装了同种类型的恶意代码防御软件的防御组内传播,既快速又避免了网络系统的额外负载,也减轻了服务器端的压力。同时用户终端计算机可以安装多个互相不冲突的反病毒软件,也就是说某一个用户终端可同时属于几个虚拟防御组。

局部的基于 Cloud-P2P 计算模型的恶意代码联合防御网络系统架构如图 2 所示。

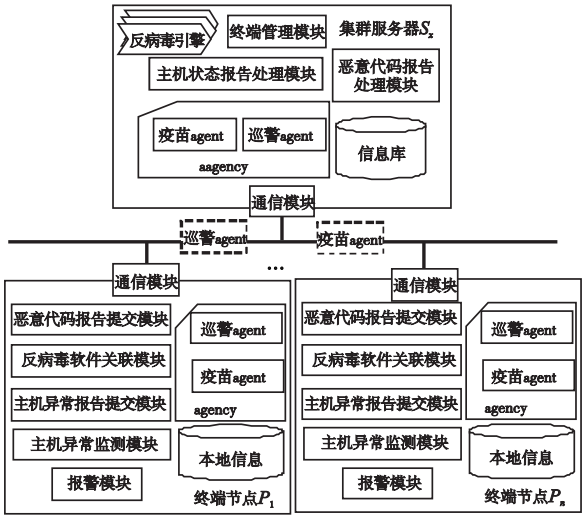

图2　系统架构(局部)

集群服务器上包括反病毒引擎、终端管理模块、主机状态报告处理模块、恶意代码报告处理模块、信息库和移动 agent(mobile agent)[12]系统模块。移动 agent 能在开放或封闭的分布式网络环境中自主地从一台主机并行或串行地迁移到其他主机,并可与当地资源主动交互地执行代码。移动 agent 包含了 agency 和 agent 主体两个部分,agency 是 agent 的运行环境。本文提出的恶意代码联合防御网络系统利用移动 agent 技术来实现承载恶意代码解决方案的疫苗(即病毒库升级包和系统补丁)agent 和能够在节点间连续迁移以检测各个节点主机防御能力、更新节点信息库的巡警 agent。

相应的,终端节点包含了恶意代码报告提交模块、反病毒软件关联模块、主机异常报告提交模块、主机异常监测模块、节点间报警模块、本地信息库和移动 agent 系统模块。这些功能模块相互联系但又各自具有比较独立的功能,既具有代表性,也具有较强的实际应用价值。反病毒软件关联模块负载可以扫描节点主机安装的反病毒软件信息,根据不同的反病毒软件提取日志信息,生成恶意代码报告及样本上报给服务器,并将已上报的病毒报告存入本地信息库。主机异常监测模块可以监测本地主机的 CPU 使用率、内存使用率、当前进程运行和网络流量等信息。用户可按一定策略设定阈值,当 CPU 和内存

使用率等超过阈值,就会发出警报提醒用户系统出现异常,并通过主机异常报告提交模块主动向服务器端告警;同时将告警信息存入本地信息库中。客户端对报告上传病毒报告、主机异常信息、用户感知报告等信息和服务器端分发的疫苗情况等形成日志存储在本地信息库中,可供用户查看。当用户发现计算机异常也可通过恶意代码报告提交模块手动地将用户感知的节点症状(如发现可疑文件信息等)发送给服务器恶意代码报告处理模块处理。恶意代码报告处理模块管理和分析恶意用户上报的恶意代码报告,根据用户报告的属性进行排序,并为报告处理模块提供接口;然后对经过初步处理和筛选的报告进行进一步处理,并根据处理的结果对提交的报告的客户端以变动信誉值的形式对提交报告的用户作出相应反馈。

2.2 网络拓扑结构

为了使集群服务器与终端之间、服务器与服务器之间、终端与终端之间能够快速进行通信,并便于疫苗 agent 和巡警 agent 在网络中的迅速迁移,合理的网络拓扑结构至关重要。

针对 Cloud-P2P 计算模型以及基于 Cloud-P2P 计算模型的恶意代码联合防御网络需求,本文提出了两种基于 DHT 的层次式网络结构 C-DHT 和 D-DHT,这两种层次式网络结构都是将 Cloud-P2P 计算网络分为由服务器节点构成的云端网层和终端网层,如图 3 所示。

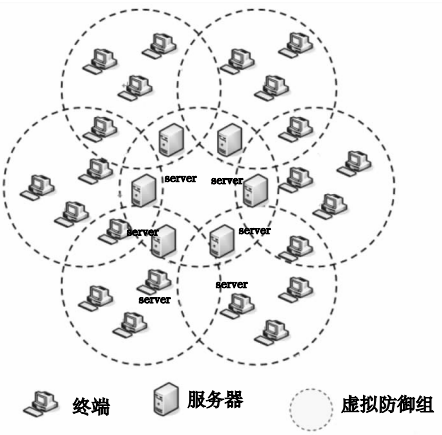

图3 层次式DHT网络拓扑结构示意图

如前所述,用户终端按照所安装的反病毒软件的不同被分为多个虚拟防御组 VDG,并与采用相同反病毒引擎的服务器交互。通过对信息资源的重组,开放的虚拟防御组可使节点更高效地相互合作,简化信息资源的存储、查找和使用,从而提高资源利用率。虚拟防御组内的终端直接与组内的其他终端进行通信和交互。

C-DHT 网络中集群服务器之间采用全连接网状拓扑,如图 4(a)所示;D-DHT 网络中服务器采用与基于 DHT 的结构化网络拓扑,如图 4(b)所示。C-DHT 和 D-DHT 网络中的虚拟防御组内节点则均采用 DHT 机制来实现组内节点之间的通信与资源共享。

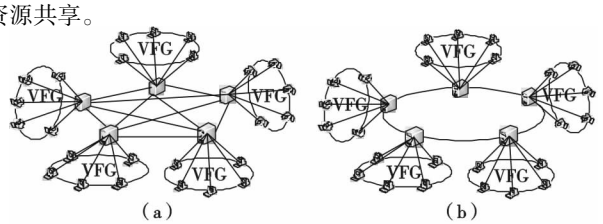

图 4 C-DHT 与 D-DHT 网络拓扑结构

这两种网络结构中,D-DHT 更加灵活,可适应动态网络变化,并具有可扩展性较好、容易管理、网络资源的利用效率高等优点。每个服务器均保存了所领导的虚拟防御组中所有终端节点的信息。子网层的每个终端节点按照 DHT 的算法同时向服务器报备完成节点加入与退出网络。C-DHT 和 D-DHT 网络系统均可参考 Chord[10,11] 算法在服务器与终端节点这两个层次构建两个环型拓扑。Chord 能在大规模网络中对节点和资源进行高效定位,并满足了负载平衡、全分布式、可伸缩性等性能要求。

当一个节点加入到 Chord 环时,必须用一个已知的节点进行引导,并对其他节点进行更新。节点加入需经过三个过程:节点加入许可过程、路由表初始化过程、更新其他节点 finger 表过程。当节点 $node_A$ 主动退出所在虚拟防御组时,首先向所在虚拟防御组服务器发送信息,要求注销其注册信息,然后 $node_A$ 通知它的前驱节点和后继节点,要退出这个网络;当前驱节点和后继节点知道收到通知后,会更新它们的数据,以维持 Chord 环的稳定。

3 网络原型系统

3.1 系统实现

本节中将详细阐述笔者领导的项目组开发的基于 Cloud-P2P 计算模型的恶意代码联合防御网络原型系统,具体包括终端、服务器端和移动 agent 三个部分,分别包含如图 2 所示的相关模块。原型系统采用 Java 语言以及 NetBeans 工具来开发,移动 agent 基于 IKV ++ 的 Grasshopper 开发。Grassoppper 符合 MASIF 和 FIPA 规范。系统实现了疫苗 agent 和巡警 agent。

疫苗 agent 实现的方法是:由服务器节点创建一个 agent,用这个 agent 读取本地疫苗(病毒库升级程序与系统补丁)文件,读进内存形成字节流;然后迁移这个 agent,当 agent 到达目的终端节点时,将内存中的文件字节流存储在节点主机;最后调用 Java 中 Runtime 类的 exec( )方法,自动执行疫苗文件或给节点主机打上系统补丁和升级病毒库。

疫苗 agent 的类图如图 5 所示,其中内容如表 1 所示。

表 1 疫苗 agent 的类说明

变量	
fileByte	用于存储疫苗文件的二进制字节
file	对应疫苗文件
arriveTime	记录 agent 到达每个节点的时间
agentState	用于描述 agent 的状态
fileLength	用于描述疫苗文件的长度
location	用于描述节点地址
version	用于描述疫苗版本信息
agentType	用于表示 agent 的类型
方法	
getVaccine( );	用于获取疫苗
getNextLocation( );	用来获得下一个节点的地址
init( );	用于初始化 agent
getName( );	得到 agent 的名称
live( );	Grasshopper 定义的 agent 的入口函数
move( );	用来使 agent 迁移
runVaccine( );	当 agent 迁移到主机时运行疫苗文件

巡警 agent 可用来监控整个网络的状况,通过在网络中各个节点中巡移,搜寻节点信息,然后返回服务器端供服务器分析处理。其类图如图 6 所示。

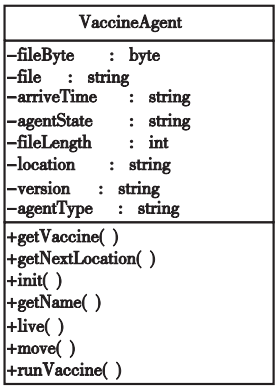

图5　疫苗agent的类图

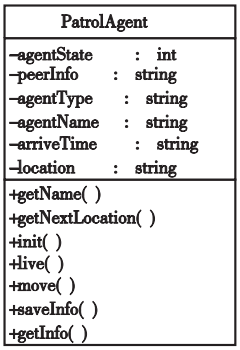

图6　巡警agent的类图

3.2　性能分析

下面对恶意代码联合防御网络系统的性能从负载等方面进行进一步分析:

a)负载均衡。随着系统功能的增强,整个网络系统增加了额外的工作。但本文将恶意代码解决方案通过服务器与终端、终端与终端两种方式来传播,从而有效减轻了服务器端的负担,负载均衡、合理。

b)反应快捷。通过大量客户端主动、及时地提供各类异常报告,使得整个网络对于新出现恶意代码的反应可以更迅速,也更准确。此外,由于恶意代码解决方案通过服务器与终端、终端与终端两种方式来传播,这样可以有效加快恶意代码解决方案的覆盖速度。

c)全面防御。单一的反病毒引擎常常在防御某一方面的恶意代码具有优势。本系统通过在服务器端集成多反病毒引擎,有效地形成优势互补,更有效地解决网络中的各类病毒。

d)兼容良好。本文提出的恶意代码联合防御网络系统是一种具有很强包容性的系统架构,可以与现有的主机/网络反病毒系统进行有机的结合,从而达到实用化的目标。

3.3　系统界面

恶意代码联合防御网络原型系统的终端主界面如图 7 所示。

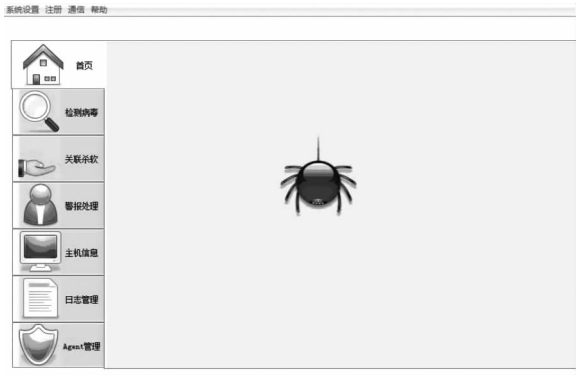

图7　恶意代码联合防御网络原型系统终端主界面

反病毒软件关联模块负责自动关联本地的反病毒软件,并读取其杀毒日志。由于反病毒软件的安装一般都会将其软件名写入注册表,并将其反病毒日志文件保存路径也写入注册表,因此可以通过读取注册表信息来得知节点安装的反病毒软件及其杀毒日志的存储路径。运行界面如图 8 所示。

图8　反病毒软件关联模块运行界面

用户可以通过恶意代码报告提交模块将病毒报告上报给服务器处理,如图 9 所示。

图9　恶意代码报告管理和提交模块界面

报警模块方便网络终端节点间、终端节点与服务器之间能够将发现的可疑恶意代码信息快速向其他节点告警,从而有效地提高了整体网络的恶意代码隔绝与预防能力,并可减轻服务器的负担,实现了网络节点的联防目标。系统运行界面如图 10 所示。

图10　报警模块界面

限于篇幅,终端系统中的主机异常报告提交模块、主机异常监测模块本文不再赘述。

相对于终端系统,服务器端系统处理各个节点上报的各种与恶意代码相关的信息,及时更新各终端节点的防御能力,从而提高整个系统的安全性。其中,恶意代码报告处理模块的管理员操作界面如图 11 所示。

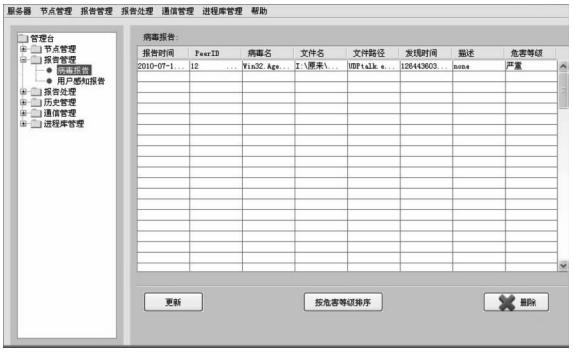

图11　恶意代码报告处理模块的管理员操作界面

由于网络中存在大量终端节点,所以服务器端收到的报告数量巨大,系统中采取依照其各个节点的信誉值作为默认的排序方式,也可选用危害等级来重新排序。其他的服务器端模块界面本文不再赘述。

(下转第 2257 页)

(上接第 2217 页)

4 结束语

本文以新型的 Cloud-P2P 计算模型为基本平台设计并构建了恶意代码联合防御网络系统,解决了现有的反病毒软件系统在应对层出不穷的恶意代码所具有的滞后性问题。恶意代码联合防御网络系统中集群服务器与用户终端群体各司其职、互通有无,组成了一个高安全防御网,协同防御恶意代码,并能够快速产生群体免疫力。后续还有很多的研究工作需要完成,包括如何激励节点主动、积极、诚实地提供恶意代码报告,以及如何实现低开销、高速度的疫苗 agent 分发机制等。

参考文献:

[1] XU Xiao-long, WANG Ru-chuan, XIAO Fu. Malicious code passive propagation model and vaccine distribution model of P2P networks [J]. Journal of Systems Engineering and Electronics, 2010, 21 (1): 161-167.

[2] 互动百科. 云安全[EB/OL]. (2009-12-20)[2010-07-28]. http:// www. hudong. com/wiki/云安全.

[3] 百度百科. 杀毒软件[EB/OL]. (2010-06-06)[2010-08-12]. http://baike. baidu. com/view/33433. htm.

[4] 陈康, 郑纬民. 云计算:系统实例与研究现状[J]. 软件学报, 2009, 20(5): 1337-1348.

[5] MILLER M. Cloud computing [M]. Beijing: Machinery Industry Press, 2009.

[6] 金海, 吴松, 廖小飞, 等. 云计算的发展与挑战[M]//2009 中国计算机科学技术发展报告. 北京: 机械工业出版社, 2010: 21-51.

[7] 徐小龙, 王汝传. 一种基于多移动 agent 的对等计算动态协作模型[J]. 计算机学报, 2008, 31(7): 1261-1266.

[8] BARROSO L A, DEAN J, HOLZLE U. Web search for a planet: the google cluster architecture[J]. IEEE Micro, 2003, 23(2): 22-28.

[9] DEAN J, GHEMAWAT S. Distributed programming with mapreduce [M]. Sebastopol: O'Reilly Media, Inc, 2007: 371-384.

[10] LUA E K, CROWCROFT J, PIAS M, *et al*. A survey and comparison of peer-to-peer overlay network schemes[J]. Communications Surveys & Tutorials, 2005, 7(2): 72-93.

[11] CORDASCO G, GARGANO L, HAMMAR M, *et al*. F-Chord: improved uniform routing on Chord[C]//Proc of the 11th Colloquium on Structural Information and Communication Complexity, Smolenice Castle. Berkeley: Springer-Verlag, 2004.

[12] MAGEDANZ T. Grasshoppera universal agent platform based on OMG MASIF and FIPA standards [EB/OL]. (2010-01-26)[2011-03-16]. http://cordis. europa. eu/infowin/acts/analysys/products/thematic/agents/ch4/ch4. htm.