基于 P2P 的蠕虫防御模型*)

周 瑛 吴中福 钟 将
(重庆大学计算机学院 重庆 400044)

摘 要 针对大规模网络环境下的蠕虫问题,本文提出一种面向蠕虫防御的层次化 P2P 重叠网模型(Hierachical Peer-to-Peer Overlay Network for Worm Prevention,简称 HPOWP)。HPOWP 通过层次式的 P2P 构架进一步提高了传统 P2P 系统的可缩放性,有效地与现有网络基础设施的拓扑结构相适应。在 HPOWP 模型中构造基于 DHT 的聚合树,提高蠕虫的识别效率。该模型为大规模复杂网络环境中的蠕虫防御提供了很好的解决方案。

关键词 蠕虫,防御,层次化,对等重叠网

Worm Defense Model Based on P2P Technology

ZHOU Ying WU Zhong-Fu ZHONG Jiang
(College of Computer Science of Chongqing University, Chongqing 400044)

Abstract To solve the problem of Worm in large-scale network environment, A hierachical Peer-to-Peer overlay network for worm prevention is proposed in this paper as HPOWP. On hierachical P2P structure, HPOWP further enhances scalability of traditional P2P system, and adapts well to the existing network infrastructure. An aggregation tree based on distributed hash tables is used for worm recognition and obtains good results. HPOWP provides a good solution for worm prevention in the large-scale and complex network environment.

Keywords Worm, Prevention, Hierachical, P2P overlay network

1 引言

随着 Internet 规模的不断扩大,网络蠕虫对计算机系统安全和网络安全的威胁日益增加,多样化的传播途径和复杂的应用环境使网络蠕虫的发生频率增高、潜伏性变强、覆盖面更广,造成的损失也更大。与传统的主机病毒相比,网络蠕虫具有更强的繁殖能力和破坏能力。

已有的蠕虫防御手段缺乏大规模网络环境下的信息共享机制。面对蠕虫的大范围传播,小范围的检测和防御手段无法进行立体防御,需要全局信息的分析才能确定蠕虫的爆发并采取相应的防御措施,因而有必要构建一个全局性的蠕虫联动防御机制,快速共享蠕虫的预警信息,达到联合遏制蠕虫传播和破坏的目的。

目前,研究在 Internet 范围内研究蠕虫的传播模式和防御技术已经成为蠕虫研究领域的共识。然而,在现有 Internet 环境下,依赖传统网络共享技术我们很难实现如此大规模的数据集中与分析。P2P 技术在分布式计算领域展示出巨大的应用前景,它为大规模蠕虫防御系统提供了有效的一个计算平台。

针对当前 P2P 重叠网络在蠕虫检测应用中存在的问题,本文提出一种面向蠕虫防御的层次化 P2P 重叠网(Hierachical Peer-to-Peer Overlay Network for Worm Prevention,简称 HPOWP)的构造方法。HPOWP 通过层次式的 P2P 构架进一步提高了传统 P2P 系统的可缩放性,有效地与现有网络基础设施的拓扑结构相适应,减少现有 P2P 网络逻辑距离与物理距离不同带来的问题,并且更加方便了各种安全措施的部署,特别是蠕虫的防御策略可以依据不同的层面采用合理的措施,更加适应实际网络环境下蠕虫的传播模式。

2 面向蠕虫防御的层次化 P2P 重叠网

2.1 网络拓扑的设计

层次化 P2P 重叠网主要解决已有 P2P 结构面临的各种问题:物理距离与逻辑距离的差距、路由效率、系统稳定性和查询优化等[1]。这也是本文构建基于 P2P 的蠕虫防御系统将要面临的挑战。采用层次化 P2P 重叠网模型作为相关研究的基础网络平台拓扑结构,充分考虑了蠕虫防御系统所面对的复杂网络环境及相关检测算法与策略的应用。面向蠕虫防御的层次化 P2P 重叠网(HPOWP)的网络拓扑图如图 1 所示。HPOWP 融合了结构化覆盖网络的规则路由能力,以支持全局蠕虫预警信息的共享和分析,以及随机覆盖网络结构中选择邻居节点的灵活性。

HPOWP 具有两层结构,下层由多个聚类网络构成。聚类网络由距离相近的参与节点通过节点聚类形成。相隔很远的节点分别参与不同的聚类网络。每一个聚类网络中根据能力选取一个节点(多个后备节点)担任该聚类的超级节点(Super Node)。各个聚类网络选择出的超级节点组织成结构化重叠网络,它们可以是任何已有的结构化重叠网络,如 Chord、CAN、Pastry 或 Tapestry 等。高层 P2P 重叠网继承了结构化重叠网络的结构特性,如规则连接、分布式哈希表空间和路由能力等。超级节点是各个聚类网络中最核心的节点,

*)基金项目:CNGI 示范工程 2005 年研究开发、产业化及应用试验项目"面向 IPv6r 的互联网安全体系结构和关键技术研究"(CNGI-04-6-2T)。
周 瑛 博士研究生,主要研究方向:信息安全、网络与网络关键技术;吴中福 教授,博士生导师,主要研究方向:远程教育、网络与网格关键技术、信息安全;钟 将 讲师,博士,主要研究方向:网络安全、免疫计算。

一般处于 Internet 体系结构的较高层次，具有高存储能力、高处理能力和高带宽，组成的高层结构化重叠网降低了网络的异质性带来的影响，使得相关应用更具可行性。

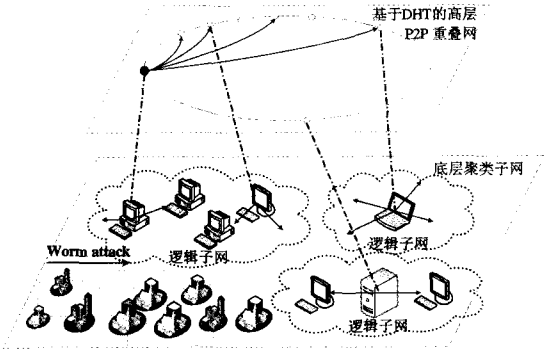

图 1　HPOWP 网络拓扑图

2.2　基于 DHT 的分布式数据聚合算法

随着 P2P 网络的不断增大，从 P2P 各个节点聚集信息变得越来越重要，例如基于 P2P 的网络存储应用、对 DHT 重叠网估算网络尺寸以及计算各个节点相关数据的统计信息（*MIN*()，*MAX*()，*COUNT*()，*SUM*()，*AVG*()）等应用。以图的方式简要描述分布式聚合计算：用一个无向图 $G(V,E)$ 表示 P2P 网络，其中，顶点集合 V 包含 n 个节点，边集合 E 指定了节点间的连接关系。在 P2P 网络中，每个节点 i 在时间段 t 拥有一个局部数据 $x_i(t)\in\beta X$，其中，$1\leqslant i\leqslant n$。对于某个给定的聚合函数 $f:X^+\rightarrow X$，分布式聚合计算就是在分布式的网络环境下算出所有局部数据的聚合结果 $g(t)=f(x_1(t),x_2(t),\cdots,x_n(t))$。

本文提出的在蠕虫防御领域的一个重要应用就是在高层 DHT 重叠网统计特定蠕虫预警信息的全局情况。这就需要从不同的超级节点将特定蠕虫的预警信息聚集到 *Key*（对信息内容进行散列的键值）所对应的节点上，如果遇到蠕虫爆发的情形，将会有大量类似的预警信息同时向一个节点汇聚，这样可能造成系统拥塞，影响预警的效果。因此，直接由各个节点汇聚信息并不合理，需要建立分布式数据聚合方法，以高效的方式统计相关蠕虫预警信息。

首先，我们对所研究的目标问题进行限制，假设这里要计算的聚合函数 $f:X^+\rightarrow X$ 是可以分解计算的，即 $f(x_1\cup x_2\cup x_3)=f(f(x_1),f(x_2),f(x_3))$，其中，$x_1,x_2,x_3\subset X$ 并且 $x_1\cap x_2\cap x_3=\phi$。前面提到了一些聚合计算，如 *MIN*()、*MAX*()、*COUNT*()、*SUM*() 和 *AVG*() 等，都是可以分解的。从而，对一个大规模数据集合的聚合计算问题可以通过局部分解计算后递归聚合，而且每次计算后输出结果的数量都会少于输入数据的数量，计算的时间和空间复杂度都将逐步递减，这对于大规模网络环境下数据的分布式聚合计算很有意义。由于我们统计蠕虫预警信息的方法也是类似的线性计算，因此该假设是成立的。

为了解决聚合问题，我们引入基于 DHT 的聚合树（Aggregation Tree Based on Distributed Hash Tables，简称 ATDHT），基于结构化重叠网的路由算法构造一个树形结构。在 ATDHT 中，每个节点对其儿子节点的数据计算 f，结果提交其父节点。通过自底向上计算 f，根节点将获得最终的全局聚合数值，而根节点的计算工作仅仅是对其儿子节点计算相关数值，计算全局聚合数值的过程如图 2 所示。

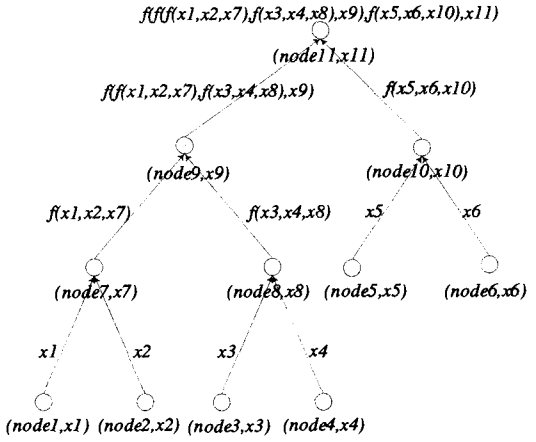

图 2　ATDHT 聚合计算示意图

其中，ATDHT 中共有 11 个节点（*node*1，*node*2，…，*node*11），各个节点分别存储着 11 个本地数据（$x1,x2,\cdots,x11$），通过逐层聚合计算，最终根节点 *node*11 获得了最终的计算结果。

2.3　CAN 中 ATDHT 的构造

本文采用 CAN 作为应用背景讨论 ATDHT 的构造，其他拓扑类型的网络都可以基于相同的思想，根据具体的拓扑构建方法有所不同。CAN 是 DHT 重叠网的代表方案之一，以它作为实例并不失一般性。

CAN 的虚平面是 d 维环空间（d-torus），每个节点负责维护邻近的超立方体空间。为叙述方便，这里讨论二维 CAN，所得结论可适用于任何维数 CAN。CAN 最主要算法的是节点路由算法，即 key 查询算法。每个节点存有 $2d$ 个邻居节点信息，如节点提出对具有 key 的内容进行查询，则分别计算 key 的 x 轴哈希函数和 y 轴哈希函数函数值作为〈key，value〉在虚平面上的坐标点，通过将查询消息不断传递给距离目标坐标点更近的邻居节点，到达负责目标坐标点所在子空间的节点，完成查询路由。依据该路由规则，我们可以方便地构造 CAN 中的一棵聚合树。对于 18 个节点的 CAN，以 N_0 为 ATDHT 根节点的路由路径如图 3 所示。

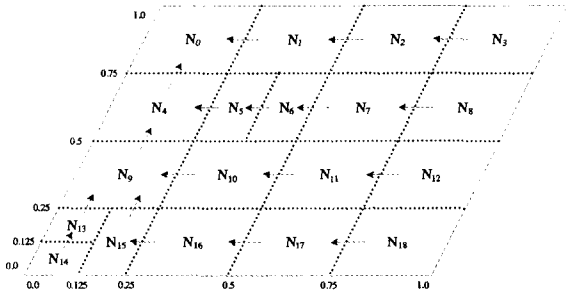

图 3　CAN 的路由图

根据聚合树的构造算法，我们构造该例中以 N_0 为根节点的聚合树如图 4 所示。

3　基于 HPOWP 的蠕虫预警机制

为了应对蠕虫带来的威胁，人们提出了各种各样防御蠕虫的方法，但目前使用的方法和策略还不能有效地阻止蠕虫的快速传播和破坏，这很大程度上归因于蠕虫扩散的规模和速度，导致现有安全防御系统难以迅速作出预警和响应。要遏制蠕虫需要解决几个关键问题：①准确识别蠕虫；②迅速预

警;③及时抑制蠕虫传播。

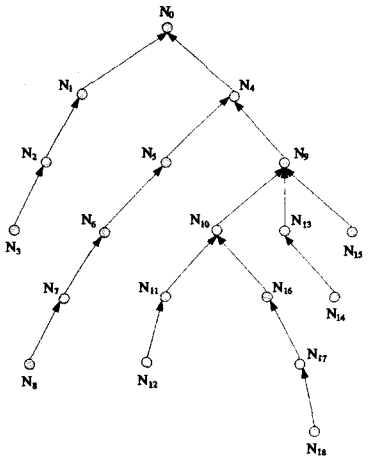

图 4 CAN 的聚合树

基于网络的关联分析和防御未知蠕虫传播行为的基本原理就是蠕虫的自相似特性[2]。主要的蠕虫相似性指标包括:被感染主机系统进程调用的相似性;大量的无效 IP 地址和无效服务请求;数据内容的相似性;节点间的传播行为具有相似性。为了验证 HPOWP 对抗蠕虫的有效性,本文采用蠕虫数据内容的自相似性来识别和阻止蠕虫传播,当然,这并非唯一选择。在 HPOWP 框架下,与蠕虫相关的相似性特征都可以作为识别算法的前提。针对不同类型的蠕虫可采用不同的特征向量,或者组合使用。

3.1 蠕虫内容的指纹识别

(1)基于内容的蠕虫特征码(蠕虫的指纹)

恶意代码(包括病毒、蠕虫、木马等)的特征,是用来进行内容匹配时使用的一组二进制序列,这个序列用来惟一地标识这种恶意代码,只要能达到这个目的,这个序列也许并不包含特别有意义的内容,就可以作为其内容特征,称为蠕虫的指纹[3,4]。正确的抽取特征码,可以使针对恶意代码的特征检测的准确率极高,商业反病毒软件对恶意代码使用特征码进行检测,一般误报率可以接近零,而漏报率在 1%左右。我们对蠕虫的特征进行详细的定义。

定义 1(连续子序列) 序列 $X=\{x_1,x_2,\cdots,x_n\}$,如果存在一个序列 $Y=\{y_1,y_2,\cdots,y_l\}$,且 $y_1=x_{m+1},y_2=x_{m+2},\cdots,y_l=x_{m+l}$,其中 $m>0,m+l\leqslant n$,则序列 $Y=\{y_1,y_2,\cdots,y_l\}$ 称为 X 的连续子序列。

定义 2(蠕虫的特征) 如果蠕虫 W 在网络上传输的数据,形成序列 $X=\{x_1,x_2,\cdots,x_n\}$,如果 X 中存在一个连续子序列 $Y=\{y_1,y_2,\cdots,y_l\}$,且其它的数据中不存在此子序列,那么我们称此子序列 Y 为该蠕虫 W 的特征。

对特征码的提取,一般手工进行,然后把各种恶意代码的特征码收集起来,则构成了恶意代码的特征库。然而蠕虫与其他恶意代码,如病毒、木马不同,它除了突然出现在极短的时间内,具有爆发性以外,还具有自相似的特性,因此使提取蠕虫的网络特征成为可能。可以看出,指纹是对一组字节执行单向函数生成的整数标识。一个好的指纹生成算法能够产生均匀分布的指纹标识,指纹不但压缩了字节长度,而且以较高的概率保证唯一性,不容易发生碰撞。

(2)Rabin-Karp 算法计算出可疑数据报文的 Rabin 指纹

一个数据报文的典型指纹可以通过计算一个 Rabin 指纹产生,方法是对报文的每 β 长度的字串计算一个 Rabin 指纹,然后选择这些指纹的一个决定性子串。

对于长度为 β 的字节序列 $t_1,t_2,t_3,\cdots,t_\beta$,可通过下面的公式计算其 Rabin 指纹:

$$RF(t_1,t_2,t_3,\cdots,t_\beta)=(t_1p^\beta+t_2p^{\beta-1}\cdots+t_{\beta-1}p+t_\beta)\bmod M \quad (1)$$

其中,p 和 M 都是整常数。

该表达式的计算形式使得每个 β 长度的 Rabin 指纹可以按照 $\{\{t_1,t_2,t_3,\cdots,t_\beta\},\{t_2,t_3,\cdots,t_{\beta+1}\},etc.\}$,的次序高效地进行计算。因此,我们对一个报文的内容从起始到末尾,以 β 为窗口大小计算 Rabin 指纹就可以利用下面的公式进行迭代:

$$RF(t_{i+1},t_{i+2},\cdots,t_{\beta+i})=(RF(t_i\cdots t_{\beta+i-1})-t_i\times p^\beta)\times p+t_{\beta+i}\bmod M \quad (2)$$

3.2 HPOWP 中蠕虫识别算法

HPOWP 对蠕虫的预警分为两个层次:①聚类子网的蠕虫特征提取与判断,通过对网络中节点可疑行为的相似性比对来初步判断蠕虫行为,并向高层重叠网提交可疑特征向量(蠕虫指纹等)。②上层 DHT 重叠网根据可疑特征向量的 Hash 值选择根节点,并构造相应的聚合树(ATDHT),收集计算全局该可疑特征向量的情况,超过指定阈值则定性为蠕虫,向全局系统发出预警信息,该算法的流程如图 5 所示。

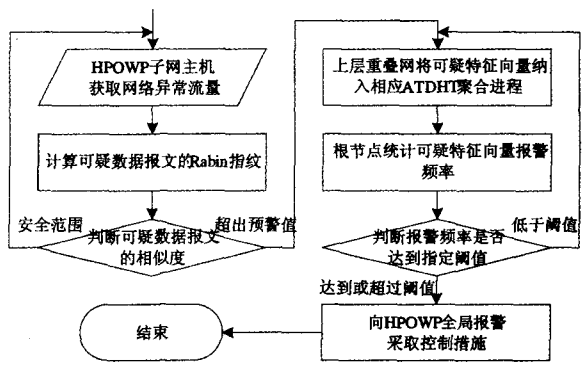

图 5 HPOWP 的蠕虫预警流程

4 实验

在本文中,仿真试验主要针对 HPOWP 本身仿真,仿真试验的目的在于验证本文提出的基于 P2P 的蠕虫预警模型的可行性,比较本文提出的预警机制与传统基于网络联动的预警机制在执行过程中,对参与网络节点的性能影响。仿真程序用 C 和 C++开发完成,运行平台为 Windows 2000, ServicePack4,CPU1.7,512M内存。在实验中,我们假定聚

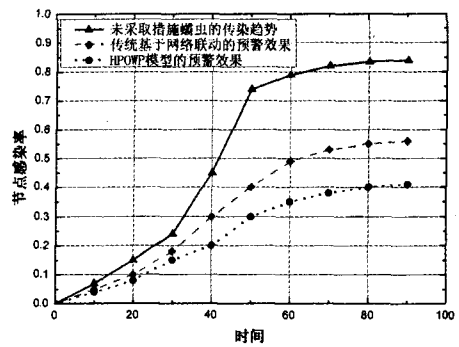

图 6 HPOWP 的蠕虫预警效果图

(下转第 99 页)

```
6  r=r->nextNode;
7}
```

该算法关于 F′的扫描代价如图 5 所示。

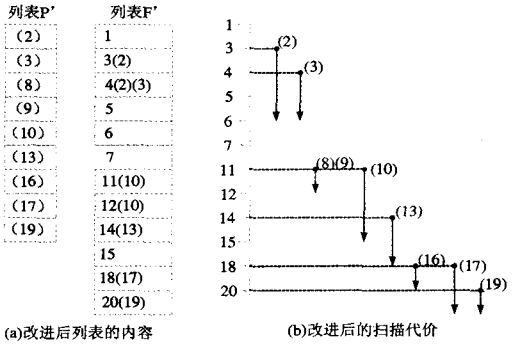

图 5　改进后的左兄弟扫描代价

从图 5(b)中可以看出，箭头的线段表示对列表 P 中的给定结点在列表 F 中的扫描范围，例如，对于列表 P′中的(2)号结点，它在列表 F′中的扫描范围是结点 3 到结点 5；(3)号结点，它在列表 F′中的扫描范围是结点 4 到结点 5；(8)号、(9)号结点，它们在列表 F′中的扫描范围都是结点 11 等等。

2.4　算法分析及结论

改进后的左兄弟/右兄弟算法中的聚集索引是按广度遍历序号建立的，所以每一个上下文结点的右兄弟的索引序号一定大于它的索引序号且是连续的，这样对每一个上下文结点进行扫描时，只需从定位的扫描始点即索引序号大于它的第一个右兄弟元素开始，扫描到第一个非它的右兄弟结点为止，即每一遍的扫描过程都只多扫描一个结点，并不需要像改进前的算法一样需扫描到右兄弟列表的末端，从而避免了大量的重复扫描。

我们通过扫描结点数量的多少来反映算法跳过不匹配元素结点的能力，评估算法性能。

在上例中，改进前对右兄弟列表总的扫描范围为 3×9+2×8+1×7+1×5+1×4+1×1=60；而改进后对右兄弟列表总的扫描范围为：

1×3+1×2+2×1+1×3+1×2+1×1+1×2+1×1=16。

改进后的扫描范围比改进前的扫描范围大大减少了。

假设右兄弟列表的结点数为 n，则改进前的扫描结点范围的数量级为 $n^2/2$，而改进后的扫描结点范围的数量级为 n，从而可以看出改进后的算法性能有很大提高。

参 考 文 献

1　Yoshikawa M, Amagasa T, Shimura T, et al. XRel: A path-based approach to storage and retrieval of XML documents using relational databases. ACM Trans on Internet Technology, 2001, 1(1):110～141

2　Jiang H, Lu H, Wang W, et al. XParent: An Efficient RDBMS-Based XML Database System. In: Proceedings of the 18th International Conference on Data Engineering, 2002

3　Grust T. Accelerating XPath location steps. In: Franklin M J, Moon B, Ailamaki A, eds. Proc. of the 2002 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Madison: ACM Press, 2002. 109～120

4　World Wide Web Consortium. XML Path Language(XPath)1.0. W3C Recommendation. 16 November 1999. http://www.w3.org/TR/xpath

5　Jiang H F, Lu H J, Wang W, et al. XR-Tree: Indexing XML data for efficient structural joins. In: Dayal U, Ramamritham K, Vijayaraman TM, eds. Proc of the 19th Int'l Conf. on Data Engineering. Los Alamitos: IEEE Press, 2003. 253～264

6　Fan C, Funderburk J, Lam Hou-in, et al. XTABLES: bridging relational technology and XML. IBM Systems Journal, 2002, 41, 4

7　刘云生，万常选，徐升华．基于关系数据库有效的实现 RPE 查询．小型微型计算机系统，2003，24(10)：1764～1771

8　Al-Khalifa S, Jagadish H V, Koudas N, et al. Structural joins: A primitive for efficient XML query pattern matching. In: Agrawal R, Dittrich K, Ngu AHH, eds. Proc. of the 18th Int'l Conf. on Data Engineering. Los Alamitos: IEEE Press, 2002. 141～152

(上接第 84 页)

类子网内节点间的通信时间远高于聚类子网间节点的通信时间，并且超级节点的处理能力、存储容量和通信速率远高于普通节点的平均通信速率。实验数据采用 Ramen 和 Nimda 的样本模拟特征。

对传统基于网络联动的预警方式和 HPOWP 的蠕虫预警模型的仿真实验结果如图 6 所示，HPOWP 模型可以更早地发现并抑制蠕虫行为。

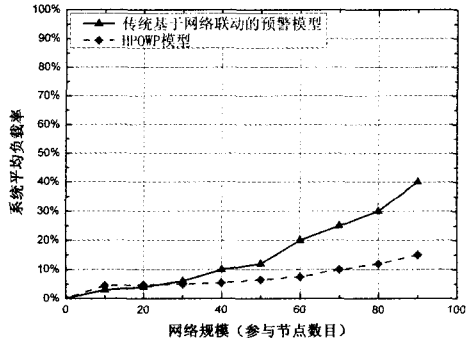

图 7　系统平均负载变化图

随着网络规模的增加，系统节点需要获取并处理更多的可疑信息，负载会不断增加。系统平均负载随网络规模变化的仿真试验结果如图 7 所示，HPOWP 模型的系统平均负载增长缓慢。

结论　本文提出的基于 P2P 的网络蠕虫预警模型(HPOWP)模型能够有效地对蠕虫行为进行预警响应。由于 HPOWP 采用了分层的 P2P 体系结构和基于 DHT 的分布式数据聚合算法，系统具有良好的伸缩性，其系统平均负载不会随网络规模的扩大而迅速增加。因此，HPOWP 更能适应大规模网络环境下的蠕虫防御要求。

参 考 文 献

1　Hathai Tanta-ngai, McAllister M. A peer-to-peer expressway over Chord [J]. Mathematical and Computer Modelling, 2006, 44: 659～677

2　卿斯汉，文伟平，蒋建春．一种基于网状关联分析的网络蠕虫预警新方法[J]．通信学报，2004，25(7)：62～70

3　Kim H, Karp B. Autograph: Toward automated, distributed worm signature detection [C]. In: Proc. of the USENIX Security, San Diego, CA, 2004. 271～286

4　Singh S, Estan C. Automated worm fingerprinting [C]. In: Proc. of the 6th ACM/USE2NIX Symposium on Operating System Design and Implementation (OSDI), San Francisco, CA, 2004. 45～60