蠕虫预警及非线性传播模型优化

佟晓筠[1] 王 嶷[2]
(哈尔滨工业大学(威海)计算机科学与技术学院 威海 264209)[1]
(哈尔滨工业大学(威海)信息科学与工程学院 威海 264209)[2]

摘 要 目前已有一些蠕虫检测系统利用蠕虫传播特性进行检测,误报率高,不能对大范围网络进行检测。为此,首先对蠕虫传播模型进行了分析和优化,提出了新蠕虫分布式传播模型。针对该模型提出了分布式蠕虫检测技术,亦即采用基于规则的检测方法监控网络蠕虫,控制台管理和协调多个检测端的工作。实验结果表明,该方法能够很好地预警蠕虫的传播行为并进行监控和报警,具有高检测率和低误报率。

关键词 蠕虫,蠕虫传播模型优化,蠕虫预警,分布式蠕虫检测
中图法分类号 TP393.08 **文献标识符** A

Worm Warning and Optimization of Nonlinear Propagation Model

TONG Xiao-jun[1] WANG Zhu[2]
(School of Computer Science, Harbin Institute of Technology, Weihai 264209, China)[1]
(College of Information Science, Harbin Institute of Technology, Weihai 264209, China)[2]

Abstract At present there are some worm intrusion detection systems which detect network worms only by using worm propagation properties and have high false alarm rate. This paper analyzed worm non-linear propagation models, realized the optimization of worm model, and proposed distributed worm propagation model. Then a distributed worm detection technology was designed according to the distributed worm propagation model. The system uses rule-based detection method to monitor network worms, and the console side manages and coordinates detection work of the client sides. The experimental results show that the technology is a good solution to worm warning and worm detection, which can give an alarm with high detection rate and low false alarm rate.

Keywords Worm, Propagation model optimization of worm, Worm warning, Distributed worm detection

1 前言

随着计算机和互联网技术的不断发展,网络安全问题日益突出。网络蠕虫攻击在各种网络安全威胁因素中位居首位。

文献[1]提出了IPv6网络中的路由蠕虫传播模型,提出了一种新型路由蠕虫RoutingWorm-v6。基于IPv6网络环境,分析了RoutingWorm-v6的扫描策略,建立了Two-Factor模型来仿真RoutingWorm-v6的传播趋势。良性蠕虫对抗恶性蠕虫的传播模型[2]表明,如果对良性蠕虫采取一些控制策略,将达到较好的抑制恶性蠕虫的效果。

针对蠕虫的大范围和快速传播已提出很多检测模型。文献[3]在分析蠕虫传播特点的基础上提出了一种使用本地网协同检测蠕虫的算法CWDMLN。该算法利用蠕虫的花瓣通信模式、无效连接以及在本地网中部署蜜罐等检测手段,协同给出蠕虫入侵的预警信息。文献[4]提出了一种蠕虫检测机制,利用很多蠕虫具有的随机扫描特性,采取丢包措施抑制其传播。虽然计算开销小,检测率高,但这种检测机制必须部署在路由器上,对环境要求太高,不适用于一般的中小型网络。

针对已有蠕虫检测模型的不足,本文对现有模型进行了优化,提出了分布式蠕虫传播模型并设计了分布式蠕虫检测系统,对于发现大范围内的蠕虫传播,减少蠕虫对网络的破坏的预警,有着很大的现实意义。

2 蠕虫非线性传播模型优化及其分布式传播模型

由于任何蠕虫传播模型的研究都是在已有传播模型的基础上进行改进的,因此下面只对几种经典的传播模型进行研究和分析,它们是Simple Epidemic Model, Kermack-Mckendrick模型和邹长春的Two-Factor模型。

2.1 SEM模型

在SEM模型[6]中,每个主机处于"易感染"状态或"被感染"状态。SEM模型假设当某个主机被感染后,它将一直处于"被感染"状态。因此,任何一个主机的状态变化只能是"易感染→被感染"或者一直处于"易感染"状态。

到稿日期:2010-07-15 返修日期:2010-11-14 本文受国家自然科学基金(60973162),山东省自然科学基金(ZR2009GM037),山东省科技攻关项目(2010GGX10132),哈尔滨工业大学(威海)校科学研究基金(HIT(WH)2009年)资助。
佟晓筠(1963-),女,博士,教授,博士生导师,主要研究方向为网络与信息安全、混沌密码学,E-mail:tong_xiaojun@163.com;王 嶷(1963-),男,硕士,教授,主要研究方向为无线传感器网络与网络安全。

$$\frac{dJ(t)}{dt}=\beta J(t)[N-J(t)] \tag{1}$$

式中，$J(t)$表示 t 时刻“被感染”结点的数目，N 是群体中的个体总数，β 是感染率。

如果选取 $N=100$，$J(0)=1$，β 分别取 0.02，0.04，0.06，根据式(1)得到一组数据，绘制出被感染节点数 $J(t)$ 关于时间 t 的函数图像，如图 1 所示。

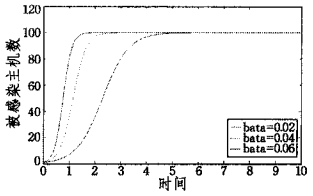

图 1　SEM 中不同参数得到的蠕虫传播趋势

SEM 模型考虑的因素较少，只考虑了易感染和被感染两种状态。这个模型只适合蠕虫传播的早期阶段，不适用于蠕虫传播的中后期。

2.2　Kermack-Mckendrick 模型

与 SEM 模型不同，Kermack-Mckendrick(KM)模型[7]考虑了主机的“易感染”、“被感染”和“免疫”3 种状态。

设 $I(t)$表示 t 时刻“被感染”主机的数量，$R(t)$表示 t 时刻“免疫”主机的数量，$J(t)$表示到 t 时刻为止被感染过的主机数量，不管 t 时刻它们是处于“被感染”状态还是“免疫”状态。于是：

$$J(t)=I(t)+R(t) \tag{2}$$

Kermack-Mckendrick 模型为：

$$\begin{cases} dJ(t)/dt=\beta I(t)[N-J(t)] \\ dR(t)/dt=\gamma I(t) \\ J(t)=I(t)+R(t)=N-S(t) \end{cases} \tag{3}$$

式中，β 为感染率，γ 为从“被感染”群体中移出变为“免疫”的比例，$S(t)$表示 t 时刻“易感染”个体的数量，N 为群体中的个体总数。

选取 $\gamma=0.03$，$N=100$，$I(0)=1$，$R(0)=1$，β 分别取 0.02，0.04，0.06，根据式(2)及式(3)绘制出被感染节点数 $I(t)$关于时间 t 的函数图像，如图 2 所示。

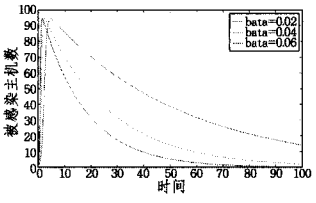

图 2　KM 模型中不同参数蠕虫传播趋势

KM 模型是对 SEM 模型的改进，考虑到了一些“被感染”主机会变为“免疫”的情况。但是 KM 模型也有不足之处。首先，KM 模型仅仅考虑了“被传染”主机的免疫，没有考虑“易感染”主机的免疫。其次，随着人们对蠕虫重视程度的不同，被感染主机的免疫率 γ 也应该是随时间递增的，所以将 γ 作为常数不太合适。

2.3　Two-Factor 模型

邹长春等人通过研究关于“Code Red”蠕虫事件的记录和文献[8]，认为上述两种模型没有考虑以下两个因素：

(1) 人为的防御措施使得一些“被感染”的计算机和“易感染”的计算机都不再处于传染循环链中，如清理被感染的主机、打补丁或升级易感染的主机，因此在建模时需要考虑易感染主机和被感染主机的免疫。

(2) 蠕虫的大规模传播造成网络拥塞和部分路由器出现问题，使得蠕虫扫描过程变慢，因此感染率应该是随时间递减的。

在 Two-Factor 模型[9]中，主机同样有 3 种状态：“易感染”状态、“被感染”状态和“免疫”状态。与 KM 模型不同，主机还可能处于“易感染→免疫”的状态转化之中，也就是说 Two-Factor 模型考虑了“易感染”主机的免疫。

Two-Factor 模型存在几个待定的动态参数：$\beta(t)$，$R(t)$和 $Q(t)$。$\beta(t)$是感染率，随时间而变化；$R(t)$表示 t 时刻从“被感染”群体中被免疫的主机数量；$Q(t)$表示 t 时刻从“易感染”群体中被免疫的主机数量。那么 t 时刻到 $t+\Delta t$ 时刻内，“易感染”主机数量变化值为：

$$S(t+\Delta t)-S(t)=-\beta(t)S(t)I(t)\Delta t-\frac{dQ(t)}{d(t)}\Delta t$$

式中，$S(t)$为 t 时刻易感染主机数目，$I(t)$表示 t 时刻“被感染”主机数目，因此有 Two-Factor 模型把易感染主机的免疫过程建模为：

$$\frac{dQ(t)}{dt}=\mu S(t)J(t) \tag{4}$$

基于 Two-Factor 模型给出的动态特性的假设，得到 Two-Factor 模型完整的微分方程组：

$$\begin{cases} dS(t)/dt=-\beta(t)S(t)I(t)-dQ(t)/dt \\ dR(t)/dt=\gamma I(t) \\ dQ(t)/dt=\mu S(t)J(t) \\ \beta(t)=\beta_0[1-I(t)/N]^{\eta} \\ N=S(t)+R(t)+I(t)+Q(t) \\ I(0)=I_0<<N;S(0)=N-I_0;R(0)=Q(0)=0 \end{cases} \tag{5}$$

式中，γ 为被感染主机的免疫率，$J(t)=I(t)+R(t)$表示被感染过的主机，μ 是常数，$\mu J(t)$表示 t 时刻易感染主机的免疫率。β_0 为感染率的初始值，指数 η 是常数，用来调节感染率对“被感染”主机数量的灵敏度。

选取 $\gamma=0.03$，$N=100$，$I(0)=1$，$R(0)=0$，$Q(0)=0$，$\mu=0.01$，$\sigma=3$，β_0 分别为 0.02，0.04，0.06。根据式(4)和式(5)绘制出被感染节点数 $I(t)$关于时间 t 的函数图像，如图 3 所示。

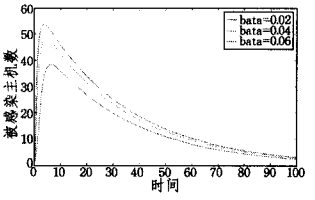

图 3　Two-Factor 模型中不同参数得到的蠕虫传播趋势

Two-Factor 传播模型是 SEM 和 KM 模型的扩展，弥补了两个模型的不足，更能适合网络蠕虫的传播状态。但 Two-Factor 传播模型没有考虑大规模自动补丁或升级对抗网络蠕虫传播的情况。同样 Two-Factor 模型将 γ 考虑成为常数。既然 Two-Factor 模型考虑到易感染主机的免疫率会随时间递增，那么被感染主机的免疫率 γ 也应该是随时间递增的，所以将 γ 作为常数不太合适。

2.4 Two-Factor 模型的优化及分布式模型

Two-Factor 模型虽然分别考虑了易感染主机以及被感染主机的免疫率，但是它将被感染主机的免疫率以及易感染主机的免疫率设置为常数，这点与实际情况不符。

从两方面考虑：

第一，已感染主机免疫率 γ。已经感染蠕虫的主机可以采用下载补丁、杀毒软件或者网络管理员人为干预的方式消除蠕虫侵害，同时自身免疫。γ 正是描述易感染主机免疫能力的参数，受许多因素的影响，比如网络中已经免疫的主机数目，官方更新的补丁是否及时以及针对发作蠕虫的杀毒软件发布等，因此 γ 必定是一个随着时间变化的值。并且在蠕虫爆发初期，γ 的值应当非常小。随着网络中免疫主机数目的增加，γ 也逐渐增大，因此需要找到一个合适的模型来描述免疫率 γ 的动态变化特性，并将其融入蠕虫传播模型中，以便更加贴切地描述实际网络环境中的蠕虫传播规律。

第二，易感染主机免疫率 μ。未被感染的主机也可以免疫，这使 Two-Factor 模型较 SEM 模型前进了一大步。但是 Two-Factor 中只考虑了 μ 的理想状态，即在蠕虫爆发传播的全过程中，μ 是一个常数。这显然与实际情况有很大出入。为了理解 μ 的动态特性，我们首先提出一个问题：易感染主机是如何获得免疫能力的呢？我们可以认为易感主机的免疫能力从两方面而来：a)从网络中其他已经免疫的主机那里得到了消息，获得了相应的补丁或者杀毒软件；b)易感染主机自身与网络监控中心取得了联系，得到了抵御蠕虫的补丁或者杀毒软件。从这两点看来，μ 在蠕虫传播的过程中必定是个变量，并且其变化规律与已感染主机免疫率 γ 类似。在蠕虫传播的初期，μ 值较小，因为网络中免疫主机数目还很少。随着越来越多的免疫主机的出现，μ 值逐渐增大，网络中易感染主机对蠕虫的抵抗能力也随之越来越强。

综合以上考虑，并通过大量实验验证模型能够较好地描述易感染主机免疫率 μ 和已感染主机免疫率 γ。

$$\gamma=\gamma_0(1+R(t)/N)^{\sigma} \tag{6}$$

$$\mu=\mu_0(1+Q(t)/N)^{\sigma} \tag{7}$$

图 4 和图 5 为 γ 和 μ 的动态变化过程。

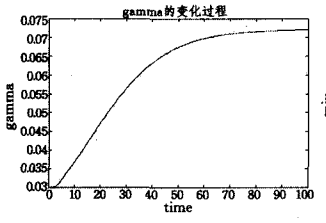

图 4 γ 的动态变化过程

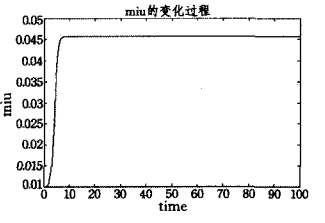

图 5 μ 的动态变化过程

通过仿真实验，优化后的模型与原 Two-Factor 模型的蠕虫传播总体趋势相同。优化前后模型如图 6、图 7 所示。

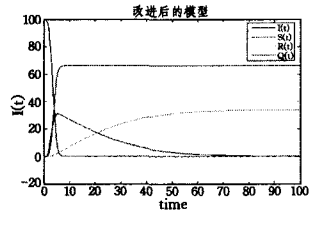

图 6 优化后 Two-Factor 模型总体趋势

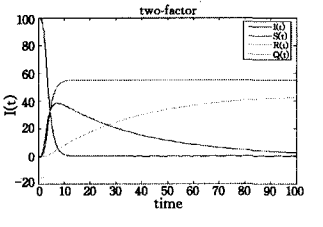

图 7 优化前 Two-Factor 模型总体趋势

不同的是，修改后的模型较原模型更快地恢复到了正常网络环境，这与实际情况吻合。

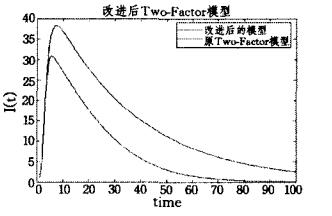

图 8 优化前后 Two-Factor 模型 $I(t)$ 性能比较

从图 8 可以看出来，改进后的模型较原 Two-Factor 模型有两点改进：

(1) 蠕虫在 100 台主机的网络环境中爆发后，在巅峰时刻感染进 38 台主机，占主机总数的 38%。而改进后的模型在巅峰时刻感染的主机数最大为 30 台左右，降低了 8 个百分点。这与原模型比较起来，更贴近实际。

(2) 改进后的模型在蠕虫爆发后的 6 天左右达到极值，然后被感染主机迅速减少。原 Two-Factor 模型则在 10 天左右达到极值且被感染主机移出缓慢。这也说明新模型能够更好地描述蠕虫传播特性。

3 分布式蠕虫预警及检测技术实现

3.1 分布式蠕虫检测模型

针对分布式蠕虫传播模型，通过对 IDS 标准框架的研究，本文设计了一个分布式蠕虫检测模型，如图 9 所示。模型属于层次结构，第一层是数据收集装置，主要负责收集网络数据包，对应 CIDF 的事件产生器；第二层是低级分析器，也就是我们常说的预处理器，进行数据包的过滤，通过预处理插件和协议分析功能，对数据包进行分片重组、代码转换、异常检测等；第三层是高级分析器，是 IDS 系统的核心，包含了对数据包内容的各项检测。本系统的检测引擎采用的是规则匹配的误用检测方法，检测经过预处理的数据包。如果发现蠕虫攻击，则产生告警。第二、三层合起来对应的是 CIDF 的事件分析器；第四层是控制台，其主要功能就是汇总各个检测端蠕虫告警、通知网络管理员做出响应，这部分对应 CIDF 的响应单元。

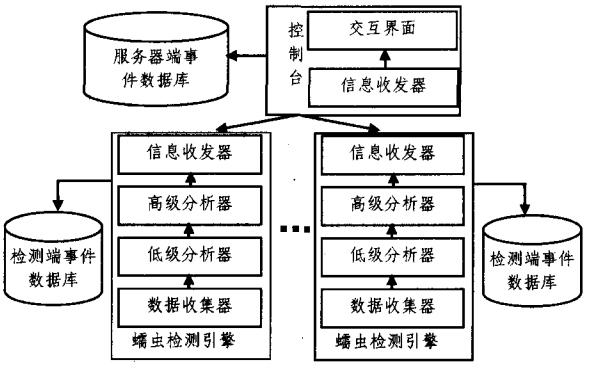

图 9 分布式蠕虫检测模型

3.2 蠕虫规则研究与设计

蠕虫规则的模板为：

动作 协议 源地址 源端口 －＞ 目的地址 目的端口 (msg:“报警信息”；content:“数据包中包括的内容”；depth:最

大搜索深度；classtype：报警类型；sid：规则编号；rev：规则版本号）

（1）Ramen 蠕虫规则

Alert tcp $HOME_NET any－＞ $EXTERNAL_NET 27374 （msg：" MISC ramen worm"；flow：to_server，established；content：" GET"；depth：8；nocase；classtype：bad-unknown；sid：514；rev：5；）

（2）CodeRed 蠕虫规则

Alert tcp $EXTERNAL_NET any －＞ $HTTP_SERVERS $HTTP_PORTS （msg：" WEB-IIS CodeRed v2 root. exe access"；flow：to_server，established；uricontent：" / root. exe"；nocase；classtype：web-application-attack；sid：1256；rev：8；）

（3）Slammer 蠕虫规则

alert udp $HOME_NET any－＞ $EXTERNAL_NET 1434 （msg：" MS-SQL Worm propagation attempt OUTBOUND"；content：" |04| "；content：" |81 F1 03 01 04 9B 81 F1| "；content：" sock"；content：" send"；classtype：misc-attack；sid：2004；rev：7；）

（4）Code Red II 蠕虫规则

Alert tcp $EXTERNAL_NET any －＞ $HTTP_NET $HTTP_PORTS （msg：" Web-IIS ISAPI. ida attempt"；uricontent：". ida?"；nocase；dsize：239；flags：A＋；）

（5）Witty 蠕虫规则

alert udp any 4000 －＞ any any （msg：" ISSPAM/Witty Worm Shellcode"；content：" |65 74 51 68 73 6f 63 6b 54 53| "；depth：246；classtype：misc-attack；sid：1000078；rev：1；）

4 实验结果及分析

4.1 实验数据

由于本系统是连接到 Internet 的，考虑到安全问题就没有使用真实的蠕虫数据包进行实验，而是采用自行开发的蠕虫模拟程序来发送符合特定蠕虫特征的数据包。主要模拟了 3 种蠕虫：Witty 蠕虫、Slammer 蠕虫和 Ramen 蠕虫。这 3 种蠕虫代表了利用 UDP 和 TCP 协议进行传播的两类蠕虫。

蠕虫模拟程序模拟蠕虫数据包的过程如下（以 Slammer 蠕虫为例）：Slammer 蠕虫将自身封装在一个大小为 376 字节的 UDP 数据包中，由任意源端口发送到网络上任意地址主机的 UDP 1434 端口；如果主机的 SQL 服务器解析服务开放，没有安装相应的补丁，蠕虫将利用缓冲区溢出漏洞对其进行感染。Slammer 蠕虫数据包第一个字节是 0x04，其中还包括 0x810xF10x030x010x040x9B0x810xF1、“sock”和“send”内容。

4.2 实验过程

（1）启动控制台，进入等待蠕虫检测端连接状态。

（2）启动蠕虫检测端并连接到服务器，进入蠕虫检测状态。

（3）启动主机 A 和主机 B，并在主机 A 和主机 B 上运行正常 TCP 和 UDP 应用程序。

（4）在主机 A 和主机 B 上运行蠕虫模拟程序，模拟 Slammer 蠕虫、Witty 蠕虫和 Ramen 蠕虫，在检测端检测到相应模拟数据包，并进行了报警，如图 10 所示。与此同时，控制台端接收到检测端告警信息，如图 11 所示。

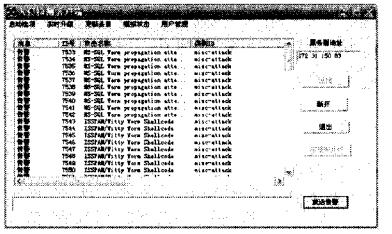

图 10 检测到 Witty 蠕虫时检测端状态

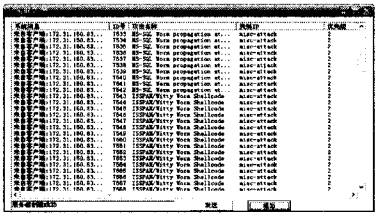

图 11 控制台接收到检测端告警

4.3 实验结果分析

按照模拟发包次数对实验数据进行列表，每次发包占一行，如表 1 所列。

表 1 实验数据列表

模拟蠕虫类型	蠕虫数据包数	报警数	检测率	漏报率
Witty	48600	41056	84.5%	15.5%
Slammer	49400	48521	98.2%	11.8%
Ramen	50000	42953	85.9%	14.1%
总计	148000	132530	89.5%	10.5%

针对表 1 对检测率、误报率和漏报率进行分析。检测率为检测到的蠕虫数据包数/蠕虫数据包总数；误报率为正常数据包误检为蠕虫数据包的数量/（正常数据包总数＋蠕虫数据包总数）；漏报率为未检测到的蠕虫数据包的数量/蠕虫数据包总数。

由于基于规则检测，没有主机模拟蠕虫流量时系统没有报警，所以误报率为 0；虽然没有百分之百的误报率为 0，但这说明了本系统的误报率接近 0。

检测率＝检测到的蠕虫数据包总数/发送蠕虫数据包总数＝89.5%

漏报率＝没有检测到的蠕虫数据包数/发送蠕虫数据包总数＝10.5%

实验表明本系统能够对各个检测网段内蠕虫事件做到大范围监控与本地检测相结合，具有低误报率、低漏报率和高检测率。

结束语 本文对蠕虫传播模型进行了深入研究并设计了优化模型，提出了蠕虫预警技术。在此基础上设计了分布式蠕虫检测系统。通过实验验证，证明了本文提出的分布式蠕虫检测技术不仅能够实现保证高检测率，也能实现对多个局域网产生的蠕虫传播信息进行综合检测。系统具有高检测率、低漏报率、误报率接近 0。

参 考 文 献

[1] 徐延贵，钱焕延，李华峰. IPv6 网络中的路由蠕虫传播模型[J]. 计算机应用研究，2009，26(10)：1-5

[2] 张殿旭，彭军，何虹. Internet 蠕虫传播模型研究[J]. 网络通讯与安全，2007：1244-1246

[3] 张新宇，卿斯汉，李琦，等. 一种基于本地网络的蠕虫协同检测方法[J]. Journal of Software，2007：412-421

[4] 赵广松，张涛. 基于蠕虫传播特性的蠕虫检测系统设计[J]. Computer Security，2009：114-118

[5] Eugene H. The Internet worm programs[J]. ACM Computer，1989，23(3)：17-57

[6] Streftaris G，Gibson G J. Statistical Inference for Stochastic Epidemic Models [A] // Proceedings of the 17th International Workshop on Statistical Modeling [C]. Chain，2002：609-616

[7] Wang Y，Wang C X. Modeling The Effects of Timing Parameters on Virus Propagation [A] // Proceedings of the ACM CCS Workshop on Rapid Malcode (WORM 2003) [C]. Washington D C：ACM Press，2003：61-66

[8] Zou C C，Gong W，Towsley D . Code Red Worm Propagation Modeling and Analysis [A] // Proceedings of the 9th ACM Symp on Computer and Communication Security [C]. Washington D C：ACM Press，2002：138-147

[9] Dantu R，Cangussu J，Yelimeli A. Dynamic control of worm propagation[J]. Information Technology，2004，1(3)：419-423

[10] Bishop M. A Model of Security Monitoring[A] // Proceedings of Fifth Annual Computer Security Applications Conference[C]. Washington DC，USA：IEEE Computer Society，1989：249-251

(上接第 76 页)

图 2 和图 3 显示了小波块阈值信道估计算法、小波软阈值估计算法和小波硬阈值估计算法的仿真性能曲线。从图中可以看出，在仿真信噪比范围内，小波块阈值估计算法的性能优于软阈值算法和硬阈值算法且几乎没有误差平层，而软阈值算法和硬阈值算法的性能基本一样且在信噪比较高时有误差平层出现。从图 2 可以看出，当 MSE 为 10^{-2} 时，小波块阈值估计算法的性能与最小二乘算法相比分别有约 7dB 的增益；与软阈值算法和硬阈值算法相比有约 2.5dB 的性能改进。图 3 显示在 SER 为 10^{-2} 时，小波块阈值估计算法与软阈值算法和硬阈值算法相比性能提升约 0.7dB。

图 4 和图 5 给出了所有设计参数不变的情况下，采用 IMT2000 Vehicular B(VB)信道进行仿真时各估计器的性能，此时信道最大时延扩展约为 224 个采样点，已经超出了系统 CP 的长度。由两张图可以看出，此时最小二乘算法和小波块阈值算法仅在高信噪比时出现了轻微的误差平层，性能基本没有恶化，但其它 3 种估计器则出现了严重的误差平层。

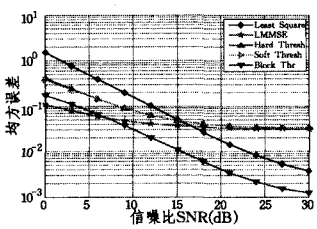

图 4 VB 信道下小波块阈值信道估计的 MSE 性能

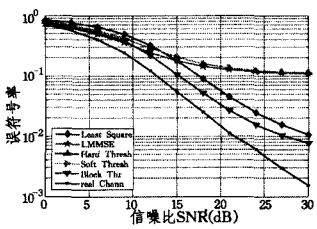

图 5 VB 信道下小波块阈值信道估计的 SER 性能

对 LMMSE 算法而言，出现误差平层原因有两个：一个原因是信道模型失配，在信道估计中 LMMSE 估计器用 IMT2000 Vehicular A 信道参数进行设计，因此在 IMT2000 Vehicular B 信道中性能会恶化；第二个原因是由信道时延扩展大于 CP 长度引入的 ICI 干扰造成的。而小波软阈值算法和小波硬阈值算法因为不能有效克服由信道时延扩展大于 CP 长度引入的 ICI 干扰而出现较大的误差平层。

从这些图中可以看出，基于小波块阈值算法的信道估计器对时延扩展有较强的鲁棒性，可以容忍时延扩展较大范围内的变化。

结束语 本文提出了 OFDM 系统信道估计的小波块阈值估计算法，该算法对插值后的 LS 信道估计值进行小波分解，并对分解后的细节系数进行小波块阈值降噪后重构出信道估计值来消除信道噪声和插值误差，从而降低了 LS 估计算法的 MSE 和 SER，取得了很好的效果。该算法不依赖任何信道和调制方式等的先验知识，且计算复杂度仅正比于数据点数。从仿真结果可以看出，该方法性能优于 LS 算法和基于小波软阈值和硬阈值的信道估计算法。在信道时延扩展超过 OFDM 系统的循环前缀长度时，该算法仍然能够保持较好的性能，因此可以采用较短的 CP 来获得同样的性能，提高系统的频谱利用率。

参 考 文 献

[1] 杨永立，朱光喜，Tassing R，等. 基于非参数平滑的 OFDM 系统信道估计算法[J]. 计算机科学，2009，36(6)：53-56

[2] 杨永立，朱光喜，陈永辉，等. OFDM 系统基于二维核回归算法的衰落信道估计[J]. 微电子学与计算机，2009(12)：104-108

[3] Ozdemir M K，Arslan H. Channel Estimation for Wireless Ofdm Systems[J]. IEEE Communications Surveys & Tutorials，2007，9(2)：31

[4] Mallat S. A Wavelet Tour of Signal Processing：The Sparse Way (3e)[M]. New York：Academic Press，2008

[5] Percival Donald B，Walden Andrew T. Wavelet Methods for Time Series Analysis. Cambridge[M]. England：Cambridge University Press，2000

[6] Tony C. On Block Thresholding in Wavelet Regression：Adaptivity，Block Size，And Threshold Level[J]. Statistica Sinica，2002，12：1241-1273

[7] Cai Tony，Low Mark. Nonparametric Function Estimation Over Shrinking Neighborhoods：Superefficiency And Adaptation[J]. The Annals of Statistics，2005，33：184-213

[8] Donoho D L，Johnstone I M. Threshold selection for wavelet shrinkage of noisy data[C] // Engineering in Medicine and Biology Society. Engineering Advances：New Opportunities for Biomedical Engineers. Proceedings of the 16th Annual International Conference of the IEEE. vol. 21，1994：A24-A25

[9] Tony C，Harrison Z. A Data-driven Block Thresholding Approach To Wavelet Estimation[J]. The Annals of Statistics，2009，37：569-595

[10] Peter H，Spiridon P，Gérard K，et al. Numerical performance of block thresholded wavelet estimators[J]. Statistics and Computing，1997，7(2)：115-124

[11] 802.16e™-2005 IEEE Std. IEEE Standard for Local and metropolitan area networks Part 16：Air Interface for Fixed and Mobile Broadband Wireless Access Systems，Amendment 2：Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands Corrigendum 1[S]. IEEE，2005

[12] ITU-R. Guidelines for Evaluation of Radio Transmission Technologies for IMT-2000. ITU RECOMMENDATION[M]. ITU-R M. 1225，1997