基于签名与数据流模式挖掘的Android恶意软件检测系统

宁 卓[1] 邵达成[2] 陈 勇[2] 孙知信[1]
(南京邮电大学现代邮政院 南京 210003)[1] (南京邮电大学物联网学院 南京 210003)[2]

摘 要 随着Android软件开发和维护的不断增多,以及恶意软件的抗检测能力逐渐增强,主流的静态检测方法开始面临一些问题:签名检测虽然检测速度快,但是对代码混淆、重打包类的恶意软件的检测能力不强;基于数据流的检测方法虽然精度高,但检测效率低。针对上述技术存在的缺点,提出了一种混合型静态检测系统。该系统改进了多级签名检测方法,通过对method与class签名进行多级匹配,提高了对代码混淆类恶意软件的检测能力。系统还改进了传统数据流分析技术,通过数据流模式挖掘,找出恶意软件频繁使用的数据流模式,省去了人工确认环节,提高了数据流分析的自动化程度与效率。两种技术的结合使得系统在检测精度与效率两方面达到一个合理的折中点。实验结果表明,该系统对于代码混淆和重打包的恶意软件具有较好的检测能力,对主流恶意软件的检测精确度达到88%。

关键词 静态分析,Android恶意软件,签名检测,数据流模式挖掘
中图法分类号 TP317 **文献标识码** A

Android Static Analysis System Based on Signature and Data Flow Pattern Mining

NING Zhuo[1] SHAO Da-cheng[2] CHEN Yong[2] SUN Zhi-xin[1]
(School of Modern of Posts, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)[1]
(Department of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)[2]

Abstract With the improvement of Android malware's resistance of being detected, traditional static analysis has faced some problems, for example, signature analysis has a high analysis speed, but it suffers repackaging and code confusion problems. Data flow analysis is preferred for its high accuracy, but it is criticized by high resources costs. To deal with the above problems, a new static analysis system was proposed by combining an improved multi-signature analysis and data flow mining method to find a balance point between the accuracy and the efficiency, in which not only the multi-signature analysis is improved by using the signatures of classes and the method, but also the frequent data flow patterns is mined in malware to avoid manual detection. The result shows the system has better capability in solving the repackaging or code confusion problem and the whole detection accuracy approaches 88%.

Keywords Static analysis, Android malware, Signature detection, Data flow pattern mining

1 引言

Android是目前最为流行的手机操作系统。根据分析机构Strategy Analytics的统计,在2014年,全球智能手机的年销售量已达到空前的13亿部,其中,Android手机的销售量在全球份额中占据绝对优势(达到81%),约10亿部[1]。另一方面,Android系统的开放性在受到应用开发者青睐的同时也带来了诸多的安全问题。当前,Android设备和Windows PC已经成为恶意软件攻击的主要对象,它们遭恶意软件攻击的次数已经趋于相近。

在这种形势下,近年来关于Android恶意软件检测技术的研究大幅度增加,主要有动态和静态两种检测技术。静态分析技术是一种在不运行代码的方式下验证代码是否满足规范性、安全性、可靠性、可维护性等指标的代码分析技术。它具有不运行代码、覆盖率高、开销相对较小的特点,因此受到了研究人员的广泛关注。

随着Android软件开发和维护的不断增多,以及恶意软件自身的抗检测能力不断增强,当前主流的静态Android恶意软件检测技术面临一些问题。比如,传统的签名检测技术在检测代码混淆或重打包的恶意软件方面效果不佳;基于Manifest配置文件的权限检测技术又时常由于"滥权"问题而产生误报;基于数据流的分析技术不仅检测耗时长,而且其检测结果需要进一步的人工确认,效率不高。

本文在总结现有Android静态检测技术的基础上,提出了一种混合型静态检测系统。该系统通过结合签名检测技术与数据流检测技术,综合改善上述技术的劣势,避免了"滥权"现象给静态检测造成的误报问题。本系统对于代码混淆、重打包的恶意软件拥有较好的检测能力,并且对一些已知的恶意软件以及使用代码混淆、重打包的恶意软件具有较快的检测速度,自动化程度较高。

图1给出了所提系统的架构图。本系统包含两大组件,即签名分析组件与数据流分析组件。签名分析组件用于生成应用的整体签名以及多级签名,并进行存储、匹配。数据流分析组件用于生成应用的数据流模式,并进行数据流模式的挖掘、存储、匹配。

本文受国家自然科学基金(61373135,61672299),南京邮电大学校级教改基金(JG01616JX73)资助。

宁 卓(1975-),女,博士,讲师,主要研究方向为入侵检测、网络行为学,E-mail:ningz@njupt.edu.cn;邵达成 男,硕士生,主要研究方向为Android安全静态分析方法。

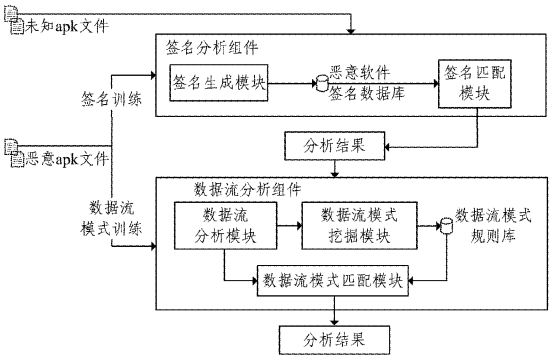

图 1 系统架构图

本系统在检测待测软件之前先进行训练，在训练阶段，系统首先选取一定的恶意软件样本分别送入签名分析组件与数据流分析组件进行签名训练与数据流模式训练。签名训练使用签名生成模块生成该批恶意软件的整体签名以及一种多级签名，并存储于恶意软件签名数据库中；数据流模式训练首先对这批软件进行数据流分析，得出每款恶意软件的数据流模式，再将这批数据流模式送入数据流模式挖掘模块进行数据流模式挖掘，将符合条件的数据流模式作为规则存储于数据流模式规则库中。

完成训练阶段后，将待检测应用送入本系统，系统首先会利用签名分析组件生成应用的整体签名与多级签名，然后在签名匹配模块中与经过训练的恶意软件签名数据库中的签名进行匹配，得出结果。如果没有规则匹配该应用的签名，则系统会将其送入数据流分析组件，该组件使用数据流分析组件对待检测应用进行数据流分析，然后与数据流模式规则库中的规则进行匹配，最终得出结论。详细的检测流程如图 2 所示。

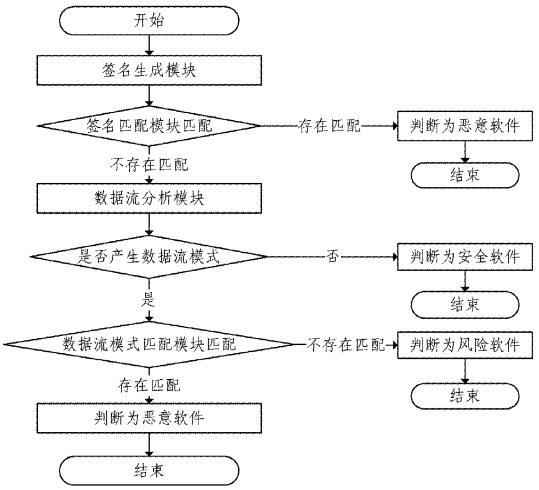

图 2 未知应用检测流程图

本文第 2 节对 Android 领域主流的静态分析方法进行总结；第 3 节阐述本系统的工作原理；第 4 节通过实验验证本系统的性能；最后总结全文。

2 相关工作

目前 Android 领域的静态分析有 3 种主流方法：基于应用权限的检测方法、数据流检测方法与签名检测。

基于应用权限的检测方法[2-4]通过检测应用申请的权限来检测应用的恶意性，这种方法开销小，但是实际上许多开发者喜欢申请过多权限以便日后的维护升级，这种“滥权”现象会使得基于权限的检测方法产生过高的误报率。

数据流检测方法[5-7]能够分析应用内部的数据流关系，追踪隐私数据的流向，从而找出隐私数据的泄露源头与泄露点。这种方法的精确度高，能够进行应用的逻辑、数据分析，但是并不能直接判断出待检测软件恶意与否，安全人员需要对数据流分析的结果进行进一步分析确认，才能给出判断。因此，该方法的检测效率低，检测速度慢，特别是对已经检测过的恶意软件，需要重新分析。

签名检测[8]的原理是比较待测应用的签名与已知恶意软件的签名是否一致，如果一致，则说明待测应用为恶意软件。传统签名检测分析速度快，但是由于它仅仅分析待检测应用的整体签名，只要攻击者对代码稍稍变动，比如改名，恶意软件的签名就会发生很大变化，从而导致传统签名检测失效。文献[9]提出了一种多级签名方法来解决上述问题，具体操作是：解压应用，扫描应用方法内部使用的各个 API 序列并排序、组合、生成 method 签名，再将各个 method 签名组合生成 class 签名，以此再生成该 apk 的整体签名。这种方法虽然能有效检测出改名的恶意软件，但是它是基于应用的 API 签名的，如果攻击者对 API 进行混淆或变动，那么这种方法将不能够进行很好的检测。本文的签名组件以此方法为基础，针对这种多级签名的劣势进行了改进。

本文的贡献如下：

1）提出了一种混合型静态检测框架，这种框架无需检测应用权限，避免了“滥权”造成的误报问题。

2）该框架使用一种改进的多级签名检测方法，提高了多级签名对基于 API 的代码混淆的恶意软件的检测能力。

3）该框架使用一种数据流模式挖掘的方法，提高了数据流分析的自动化程度与效率。

3 设计与实现

本系统包含两大组件：签名分析组件与数据流分析组件。签名组件用来检测已知恶意软件与使用代码混淆、重打包以企图躲避检测的恶意软件，数据流分析组件用于检测签名检测可能遗漏的未知恶意软件。

3.1 签名分析组件

签名分析组件包含签名生成模块、恶意软件签名数据库以及签名匹配模块，用于完成对恶意软件的签名生成、存储以及待检测应用的签名生成、匹配功能。

签名生成模块是签名分析组件的核心，它能够生成两种类型的签名：1）以 apk 包为基础的整体签名，用于检测已知的恶意软件，加快重复恶意软件的检测速度；2）通过解压 apk 包，扫描应用使用的 API 函数而生成的基于 API 的多级签名，这种签名用于检测恶意代码逻辑已知但由于重打包或代码混淆而不能被传统签名方法所检测的未知恶意软件。

文献[9]存在的主要问题是：由于该多级签名方法仅仅比较最后生成的 apk 级签名，因此一旦攻击者使用基于 API 的混淆方式，便可造成最终的 apk 级签名不一致，从而躲避检测。

这种 API 的混淆主要有 3 种形式：在恶意代码所在类的外部插入混淆用的 API，在恶意代码所在类的内部、恶意代码所在方法的外部插入混淆 API，以及在恶意代码所在的方法的内部插入混淆 API。针对前两种情况，本文对多级签名的改进措施是生成待检测应用的类签名与方法签名，将待检测应用与恶意软件签名数据库中的类签名与方法签名进行匹配，取代原先仅匹配 apk 级签名的做法，从而得出结论。

本文的多级签名生成方法如图 3 所示。

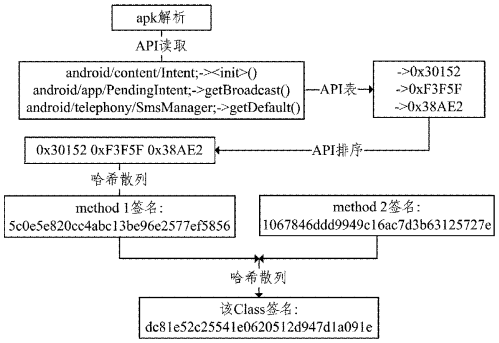

图 3　多级签名的生成流程

在解压 apk 之后,遍历字节码文件中的 API 函数,首先根据预先设定的 API 表查找各个 API 对应的 16 进制数值;然后根据 API 进行排序组合,得到该方法的 API 调用 ID 序列,该序列由若干个 API 对应的 16 进制数值组成;最后将其一并散列成签名,这样就能得到该方法的签名;最后,对该类的所有方法签名进行排序、组合、散列,得到该类的签名。

为了更好地阐述本文多级签名检测的原理,使用一款 ADRD 类的恶意软件进行说明,如表 1 所列。该款恶意软件使用 5 种混淆方式进行代码混淆,对这 5 种方式的混淆分别生成 apk 的整体签名与多级签名。

表 1　一款 ADRD 类恶意软件的整体签名与多级签名

类别	APK 整体签名	APK 多级签名(APK 级签名)	恶意代码所在 CLASS 级签名	恶意代码所在 METHOD 级签名	混淆方式
ADRD	1fc775eb	90282A7E	65E3CA04	A63E8B46	改名混淆
	acb4dd25	9319C03F	E815302D	E08F0024	
	87d995b1	5933CDE5	F7580896	45607AA3	
	73accdcf	A0B70B3C	CECC7854	F73A27C4	
ADRD	BCA53C82	90282A7E	65E3CA04	A63E8B46	插入无效 method
	4D7CDCA6	9319C03F	E815302D	E08F0024	
	09003ABB	5933CDE5	F7580896	45607AA3	
	C591F4E6	A0B70B3C	CECC7854	F73A27C4	
ADRD	0909EB77	B4C07F2B	65E3CA04	A63E8B46	类外插入混淆 API
	0406898B	32833256	E815302D	E08F0024	
	3A08D507	6B0D86AE	F7580896	45607AA3	
	7A374637	FA84955A	CECC7854	F73A27C4	
ADRD	43B87614	838C0D51	AFB6B692	A63E8B46	方法外部、类的内部插入混淆 API
	6ADA24E7	4A30B492	D3FBEFE5	E08F0024	
	0BE6473A	B5F24245	7B9526F6	45607AA3	
	2877BDD8	E002BD82	B569849D	F73A27C4	
ADRD	ED6E1FD1	A7186D80	25ABE11F	12BE9F0A	方法内部插入混淆 API
	ADE41C02	1B11FEF9	24F4F859	1D6C0B0F	
	E8CCAAC0	65F973EF	407AB610	B1DD2FFD	
	2796B4BF	6E1EE152	BA643E37	3438A296	

从表 1 中可以看出,主流多级签名可以抵御改名混淆与插入无效 method 的混淆方式,这两种混淆方式会改变 apk 整体签名,但并不会影响多级签名的 apk 级签名。对于后 3 类基于 API 的混淆方式,仅检测 apk 级签名是无效的,这 3 种方式均会使得 apk 级签名发生改变。但是可以看到,在类外插入混淆 API 时其恶意代码所在的 class 级签名不变;在恶意代码所在的方法外部、类的内部插入混淆 API 时,恶意代码所在的 method 级签名不变。因此,本文认为,通过检测 apk 的 class 级签名与 method 级签名,可以改善多级签名对基于 API 混淆的恶意软件的检测能力。对于最后一类 API 的混淆方式,签名分析组件并不能进行有效检测,多级签名均发生了变化,因此需要使用数据流分析组件进行数据流模式的分析。

通过签名生成模块,可以得到应用的整体签名以及对应的类签名与方法签名。在系统的训练阶段,将恶意软件的整体签名、类签名、方法签名进行存储;在系统的检测阶段,生成待检测应用的整体签名以及这种多级签名,并将这些签名与恶意软件签名数据库中的签名进行匹配,若待检测应用的某个类文件或者某个方法的签名与数据库中的一致,则判断待测软件为恶意软件。

3.2　数据流分析组件

签名分析组件并不能有效检测第三种基于 API 的代码混淆恶意软件,这是由于签名技术归根到底只是一种字符的识别与匹配技术,它并不能对应用软件进行深度分析,无法梳理出 API 之间的数据流关系,因此总是存在局限性。为了能够检测第三种基于 API 的代码混淆恶意软件,还需要经过一定的数据流分析,而且这种数据流分析应该尽可能高效。

主流的数据流分析方法[5-7]需要对数据流分析结果进行人工确认,这大大降低了数据流的分析效率。而通过挖掘恶意软件常用的频繁数据流模式并以此作为检测标准则可以提升效率。

数据流分析组件使用数据流模式挖掘方法,通过对大量恶意软件的数据流进行分析,得到恶意软件频繁使用的数据流模式数据流模式是一组二元关系对,形如(source, sink)。source 为隐私数据的泄露源,通常是一组能够读取用户隐私的 API 函数,例如 getDeviceId()。sink 为隐私数据的泄露点,通常是一组能够发送数据的 API 函数,例如 sendTextMessage()。在大多数的应用环境中,一组数据流模式代表了一条可能的隐私数据泄露路径。

通过对待检测应用进行数据流分析得出数据流模式,再将其与数据流模式规则库中的恶意软件频繁使用的数据流模式进行匹配,如果发现一致的数据流模式,就判断为恶意软件;反之,则判断为安全软件或者风险软件,如图 2 所示。

本文使用数据流分析工具分析了大量从 http://contagiominidump.blogspot.com/下载的恶意软件,这些恶意软件会泄露用户隐私。本文根据这些泄露隐私的数据流模式的出现频率,分别计算了每种规则的 $P(source\ sink)$ 以及 $P(sink\ |$

source),即数据流模式(*source*,*sink*)出现的联合概率与条件概率,通过对两种概率的筛选,挖掘出了一些恶意软件频繁使用的数据流模式规则并将其依照类别进行归类,部分规则如表 2 所列。

表 2 恶意软件频繁使用的部分数据流模式示例

类别	Source	Sink
隐私窃取,通过短信发送	TelephonyManager. getDeviceId() TelephonyManager. getSimSerialNumber() TelephonyManager. getLine1Number()	sendTextMessage()
隐私窃取,通过网络发送	TelephonyManager. getDeviceId() TelephonyManager. getSimSerialNumber() TelephonyManager. getLine1Number()	URL. openConnection() HttpClient. execute() URL. openStream()

4 实验

4.1 签名分析模块实验

本实验采取的数据集来源于文献[10]的恶意软件数据集,选取了其中的 30 款恶意软件。这些恶意软件分别来自于 3 个恶意软件类别,分别是 ADRD,Kungfu 和 BaseBird。再将 30 款应用分别进行 3 类基于 API 的代码混淆与 2 类普通的代码混淆,得出 150 款变种恶意软件,利用这 150 款恶意软件进行测试。部分实验结果如表 3 所列。

表 3 签名分析模块对 150 款变种恶意软件的实验结果

类别	样本数量	使用原多级签名方法检测到的样本	使用本文多级签名方法检测到的样本
ADRD	50	20	40
Kungfu	50	20	40
BaseBird	50	20	40

实验结果基本达到预期,原多级签名方法可以有效检测 2 类普通代码混淆(即改名混淆与插入无效 method),但是无法检测基于 API 的代码混淆恶意软件;本文的多级签名方法提高了多级签名对基于 API 的代码混淆恶意软件的检测能力,但仍不能检测部分恶意软件,这是由于通过使用第三类的 API 混淆方式导致签名组件检测失效。

4.2 数据流分析模块实验

由于签名组件不能有效检测使用第三类 API 混淆的恶意软件,需要使用数据流分析组件进行检测,因此本系统对混淆恶意软件的检测精度最终取决于数据流分析组件的检测精度。

本文将恶意软件集[10]中未被使用的恶意软件作为测试样本,以检验数据流分析组件的准确率,验证数据流模式规则的有效性。在其中选取了常见的 8 个类别共计 200 种恶意软件进行实验。实验结果如表 4 所列。

表 4 数据流分析模块对 200 款恶意软件的实验结果

类别	样本数量	检测到的样本	检测率
Adrd	80	68	0.85
Boxer	2	2	1
Cosha	3	2	0.67
CrWind	5	5	1
FakePlayer	6	0	0
Genimi	60	57	0.95
HipoSMS	4	0	0
Kungfu	40	38	0.95
总计	200	172	0.86

从表 4 可知,检测成功率较高的类别有 Adrd,Genimi 和 Kungfu 等,它们是通过后台进行隐私窃取的常见恶意软件类别。为了完成隐私数据的窃取,这些类别的恶意软件通常会使用系统 API 来进行隐私数据的读取与发送,例如,读取被感染手机的联系人信息,用网络将其发送至接收端。而对于检测率较低的类别,诸如 HipoSMS,它们使用常量作为参数,向指定号码发送短信,或者拦截某一号码的短信,因此并不能很好地被数据流检测方法检测。

对比目前主流的数据流分析工具 FlowDroid,本文的数据流分析组件使用经过数据流模式挖掘的数据流规则替代人工确认,提高了数据流检测的效率。

数据流分析组件对使用读取与发送隐私数据相关 API 的恶意软件有良好的检测效果,而一些使用常量作为参数的 API 并不能被数据流检测方法有效地检测出来,因此这部分恶意软件能够躲避本系统的数据流检测。

4.3 混合实验

由于真实的检测环境中,未知恶意软件通常由变种的已知恶意软件以及完全未知的恶意软件构成,为了综合计算系统的最终检测精度,本文重新随机选取了数据集[10]中的 30 款已知恶意软件并对它们进行变种,按比例选取其中 100 款作为实验数据集的变种恶意软件测试类。在数据集[10]中重新按照恶意软件类别比例选取了 300 款完全未知的恶意软件来作为完全未知恶意软件测试类。实验结果如表 5 所列。

表 5 本系统对 400 款未知恶意软件的实验结果

未知恶意软件类别	样本数量	本系统检测到的样本数量	检测率
变种恶意软件	100	94	0.94
完全未知恶意软件	300	259	0.86
总计	400	353	0.88

从表 5 可知,通过联合数据流分析模块,本系统检测变种恶意软件时比单独使用签名模块的检测精度有所提高。

根据上述实验,本系统最终的检测率为 88%。在检测速度方面,本系统由于使用了签名匹配与数据流模式匹配的匹配方法,在检测一些已知的或者变种的恶意软件时其平均速度约为 2s,在检测未知恶意软件时其平均速度约为 16s。

结束语 本文提出了一种混合型静态检测系统,以平衡静态分析的效率与精度,综合改善对使用 API 混淆的恶意软件的检测效果。本系统使用了一种改进型的多级签名技术,提高了多级签名对基于 API 的代码混淆的恶意软件的检测能力。本系统还使用了一种数据流模式挖掘的方法来提高数据流分析的自动化程度与效率,使得数据流分析不再需要进行人工确认。实验表明,本文的系统能够有效检测两类基于 API 的代码混淆恶意软件,最终的检测成功率为 88%。在未来的工作中,将加强系统对动态加载恶意代码的应用的检测能力;同时,通过改善数据流的分析策略,提高数据流分析的精确度。

参 考 文 献

[1] ANALYTICS S. Android shipped 1 billion smartphones world-

wide in 2014[OL]. http://www. strategyanalytics. com/default. aspx?mod=reportabstractviewer&a0=10539.

[2] ENCK W, ONGTANG M, MCDANIEL P. On lightweight mobile phone application certification[C]// Proc. of the 16th ACM Conf. on Computer and Communications Security (CCS 2009). 2009:235-245.

[3] LIU X, LIU J Q. A Two-layered Permission-based Android Malware Detection Scheme[C]// 2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering. 2014:142-148.

[4] GUO C K, XU J, LIU L, et al. MalDetector-Using Permission Combinations to Evaluate Malicious Features of Android App [C]// Software Engineering and Service Science (ICSESS). 2015:157-160

[5] ZHAO Z B, OSONO F C C. Trustdroid: Preventing the use of smartphones for information leaking in corporate networks through the used of static analysis taint tracking[C]// 7th International Conference on Malicious and Unwanted Software (MALWARE). 2012:135-143

[6] FRITZ C, ARZT S, RASTHOFER S, et al. Highly Precise Taint Analysis for Android Applications[J]. Cs. ucdavis. edu, 2013, 3(2):151-157.

[7] KLIEBER W, FLYNN L, BHOSALE A, et al. Android taint flow analysis for app sets [C]// ACM Sigplan International Workshop on the State of the Art in Java Program Analysis. 2014:1-6

[8] QIN Z Y, YANG Z Y, DI Y X, et al. Detecting repackaged android applications [J]. Lecture Notes in Electrical Engineering, 2014, 277:1099-1107.

[9] ZHENG M, SUN M S, LIU C S. Droid Analytics: A Signature Based Analytic System to Collect, Extract, Analyze and Associate Android Malware[C]// proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. Melbourne, VIC, Australia, 2013: 163-171.

[10] ZHOU Y, JIANG X. Dissecting Android Malware: Characterization and Evolutionin [C]// Proceedings of 33rd IEEE Symposium on Security and Privacy. 2012:95-109.

(上接第 285 页)

显然,LSDC 算法能更好地提高网络的稳定性,延长网络的生命周期。

结束语　本文提出一种射频能量捕获无线传感器网络分层分步数据收集算法,对网络进行分层,并以分步的方式逐层地将节点中的数据全部收集到 Sink 节点,选择能量状况最佳的节点作为传递节点。本算法从节能和能量消耗均衡两个方面来提高网络的稳定性,延长网络的生命周期。理论分析和仿真实验都充分证明了这一点。

参 考 文 献

[1] LU X, WANG P, NIYATO D, et al. Wireless Networks With RF Energy Harvesting: A Contemporary Survey[J]. IEEE Communications Surveys & Tutorials, 2015, 17(2):757-789.

[2] Riquelme J A, SOTO F, SUARDAZ J, et al. Wireless sensor networks for precision horticulture in southern spain [J]. Computers and Electronics in Agriculture, 2009, 68(1):25-35.

[3] NADERI M Y, CHOWDHURY K R, BASAGNI S. Wireless sensor networks with RF energy harvesting: energy models and analysis [C]// IEEE Wireless Communications and Networking Conference. New York: IEEE Press, 2015:1494-1499.

[4] FLINT I, LU X, PRIVAULT N, et al. Performance analysis of ambient RF energy harvesting: A stochastic geometry approach [C]// Proc of the Global Communications Conf. New York: IEEE Press, 2014:1448-1452.

[5] TIAN Y L, CHENG P, HE L, et al. Optimal reader location for collision-free communication in WRSN[C]// Proc of the Global Communications Conf.. New York: IEEE Press, 2014:4418-4423.

[6] LI T, FAN P Y, CHEN Z C, et al. Optimum transmission policies for energy harvesting sensor networks powered by a mobile control center [J]. IEEE Transactions on Wireless Communications, 2016, 15(7):6132-6145.

[7] EROL-KANTARCI M, MOUFTAH H T. Mission-aware placement of RF-based power transmitters in wireless sensor networks [C]// Proc. of the Computers and Communications. New York: IEEE Press, 2012:12-17.

[8] CHEN X, HE C, JIANG L G. The tradeoff between transmission cost and network lifetime of data gathering tree in wireless sensor networks [C]// Proc. of the Communications. New York: IEEE Press, 2013:1790-1794.

[9] LI H J, JAGGI N, SIKDAR B. Relay scheduling for cooperative communications in sensor networks with energy harvesting [J]. IEEE Trans on Wireless Communications, 2011, 10(9):2918-2928.

[10] BAO X C, DING G Q. An routing algorithm for maximizing network performace in energy harvesting wireless sensor network [C]// International Conference on Information Science and Control Engineering. New York: IEEE Press, 2016:1267-1270.

[11] IMON S K A, KHAN A, DI FRANCESCO M, et al. Energy-Efficient randomized switching for maximizing lifetime in tree-based wireless sensor networks [J]. IEEE/ACM Trans on Networking, 2015, 23(5):1401-1415.

[12] MARTINEZ G, SHUFANG L, CHI Z. Multi-commodity online maximum lifetime utility routing for energy-harvesting wireless sensor networks [C]// Proc. of the Global Communications Conf.. New York: IEEE Press, 2014:106-111.

[13] YAO R N, WANG W, SOHRABY K, et al. A weight-optimized source rate optimization approach in energy harvesting wireless sensor networks [C]// IEEE Global Communications Conference. New York: IEEE Press, 2012:1789-1793.

[14] SEAH W K G, OLDS J P. Data delivery scheme for wireless sensor network powered by RF energy harvesting [C]// Proc. of the Wireless Communications and Networking Conf.. New York: IEEE Press, 2013:1498-1503.

[15] BALANIS C A. Antenna Theory: Analysis and Design[M]. John Wiley & Sons, 2012.

[16] HEINZELMAN W B, CHANDRAKASAN A P, BALAKRISHNAN H. An application-specific protocol architecture for wireless microsensor networks [J]. IEEE Trans. on Wireless Communications, 2002, 1(4):660-670.