

# 无线传感器网络重编程技术研究<sup>\*</sup>

张羽 蒋泽军 周兴社

(西北工业大学计算机学院 西安 710072)

**摘要** 有效和可靠的无线传感器网络重编程技术是解决大规模传感器网络难于管理和维护的重要途径。介绍了无线传感器网络重编程的概念和基本组件,提出了无线传感器网络重编程技术面临的挑战和性能评价标准,分析了当前的研究进展并进行了比较,最后指出了无线传感器网络重编程技术的发展趋势。

**关键词** 传感器网络,网络重编程,代码分发,动态重配置

## Research on Reprogramming Wireless Sensor Networks

ZHANG Yu JIANG Ze-jun ZHOU Xing-she

(School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China)

**Abstract** An efficient and reliable reprogramming service is an important approach of management and maintenance of wireless sensor networks. In this article, the concept and basic components of reprogramming wireless sensor networks are introduced, and the technical challenges and characteristics of reprogramming wireless sensor networks are proposed. The current research progresses are analyzed and compared. Finally, the open research problems in this area are also pointed out.

**Keywords** Sensor networks, Network reprogramming, Code dissemination, Dynamic reconfiguration

## 1 引言

无线传感器网络(Wireless Sensor Networks, 简称 WSNs)作为一种全新的信息获取和处理技术已经成为国内外学术界和工业界关注的热点<sup>[1,2]</sup>。WSNs是由大量、低成本、集成计算和通信能力、有限能量的微型传感器节点以 Ad Hoc 方式构成的分布式系统。

由于受周围物理环境和自身资源极端限制的影响,WSNs表现出高度不确定性和动态性的特征<sup>[3]</sup>,使得管理和维护运行在恶劣远程环境中的 WSNs 面临着诸多挑战。WSNs 重编程技术是一种有效的解决途径<sup>[4]</sup>。此外,为了支持多样的感知应用,共享已部署的网络硬件资源和自适应不可预见的事件,也使研究 WSNs 重编程技术具有重要的意义<sup>[5,6]</sup>。

WSNs 重编程是在传感器网络首次部署完成后对其进行远程任务再分配、节点软件更新和网络功能重配置的过程<sup>[4,6]</sup>。从重编程的范围来说,可能是网络中所有节点,也可能是部分指定节点;从重编程的粒度来说,可能是完整的程序映像或程序的部分模块,也可能是虚拟机脚本。

本文将全面介绍和讨论有关 WSNs 重编程技术方面的研究进展。在第 2 节介绍一个典型的 WSNs 重编程概念模型并说明 WSNs 重编程服务的主要组件和基本过程;第 3 节提出了 WSNs 重编程技术的一系列挑战和相关的性能评价标准;第 4 节讨论了目前在 WSNs 重编程技术上具有代表性的研究项目,并做出了相应的比较;最后是文章的总结。

## 2 无线传感器网络重编程模型

图 1 所示为一个典型的 WSNs 重编程概念模型,包括网络管理员使用的用户界面 UI、更新软件版本数据库、代码优

化组件、代码分发协议和节点执行环境。其中,代码优化组件使要分发的更新程序的代码量最小化,代码分发协议完成可靠的软件分发服务;节点执行环境用来切换和执行新更新的程序。

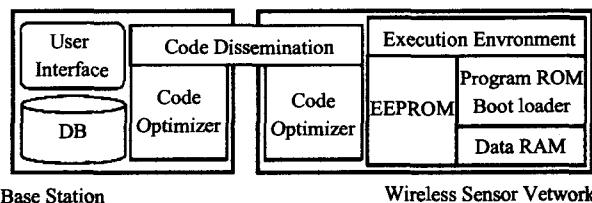


图 1 WSNs 重编程模型

通常的 WSNs 重编程过程是一个网络管理员使用用户 UI 选择重编程策略,如软件更新范围或更新粒度,确定软件更新计划。UI 从更新软件版本数据库中取出软件的新旧版本将它们传递给代码优化组件进行不同版本的比较,并生成压缩后的更新代码。此后,更新代码以数据包的形式由代码分发协议传送到全网的所有节点或指定的节点中。当更新代码数据包到达目标节点时,完成程序代码的重构,最后节点执行环境通过重新启动系统完成新旧程序的切换。

## 3 无线传感器网络重编程挑战

### 3.1 无线传感器网络重编程挑战

不像传统分布式系统中所使用的重编程工具(如 IBM 的 Tivoli Configuration Manager<sup>[7]</sup>, HP 的 OpenView<sup>[8]</sup> 和微软的 SMS<sup>[9]</sup>)那样,可以不考虑运行平台的资源问题,WSNs 重编程服务必须运行在资源极端受限的传感器节点环境之中,这给我们带来了一系列新的挑战:

<sup>\*</sup>国家自然科学基金资助项目(编号:60573161)。张羽 博士研究生,CCF 会员,主要研究方向为传感器网络重编程和操作系统;蒋泽军教授,主要研究方向为嵌入式系统和网络安全;周兴社 教授,博士生导师,主要研究方向为普适计算、分布实时计算和嵌入式网络计算。

- 存储容量有限。以 MicaZ 和 Toles 平台为例, 它们的 Data RAM 的容量仅有 4k 字节。WSNs 重编程算法的时空复杂性应该符合传感器节点存储资源的要求。

- 电源能量有限。在计算、读写、感知和收发数据等各种操作中, 发送数据包和向外部闪存 EEPROM 写数据块是高耗能的<sup>[10]</sup>, 因此在 WSNs 重编程设计时要尽可能地避免。

- 网络伸缩性强。可伸缩性是大规模传感器网络设计的关键<sup>[11]</sup>, WSNs 重编程算法应该既能适应网络节点数目的增长, 也能适应网络节点密度的变化。

- 网络高度动态。WSNs 由于是通过无线广播介质进行通信, 因此会受到无线通信固有的不可靠性影响, 同时网络中的节点具有移动性, 并且经常出现新节点的加入和已有节点的失效, 导致 WSNs 的拓扑在不断变化, 这对 WSNs 重编程技术提出了更高的要求。

- 广播风暴和隐藏终端问题。泛洪是 WSNs 实现代码分发的一种最直接方法, 但是会产生严重的广播冗余、竞争和冲突<sup>[12]</sup>, 同时无线通信中的隐藏终端问题也会引起接收节点处的数据包冲突、损坏和丢失, 致使网络性能急剧下降, 因此 WSNs 重编程服务必须要解决这些问题。

- 执行环境能力有限。由于大部分传感器节点的硬件不具备 MMU, WSNs 节点的操作系统与其上的程序没有明确的边界, 如 TinyOS<sup>[13]</sup> 与应用程序组件是在编译时地址绑定, 这使操作系统代码与应用程序代码成为一个单一的可执行映像, WSNs 重编程模块不能独立加载, 尽管 SOS<sup>[14]</sup>, Contiki<sup>[15]</sup> 等支持动态模块加载, 但执行环境的灵活性与更新代价之间的权衡仍是挑战。

### 3.2 传感器网络重编程性能评价标准

WSNs 重编程的性能直接影响其有效性和可用性, 对能否完成大规模 WSNs 的管理和维护起着非常重要的作用。下面, 我们给出几个评价 WSNs 重编程性能的标准。

- 传输可靠性。WSNs 重编程中的代码分发可以看成是一种特殊的数据分发过程, 两者的主要区别之一在于前者必须使所传输的每个代码字节都能正确地达到需要更新的目标节点。

- 操作自治性。WSNs 重编程应该尽可能减少人为的干预, 具备从网络其他节点中自主获取更新代码的能力。

- 更新实时性。WSNs 重编程过程应该尽可能地缩短代码更新的时间, 甚至需要满足一定的实时性要求, 一个好的 WSNs 重编程服务应该对感知应用的影响最小化。

- 更新安全性。WSNs 重编程安全机制应该提供一定级别的完整性、私密性和认证服务, 确保远程代码更新的安全完成。

- 低存储使用。WSNs 重编程过程应该尽可能减少对存储资源的使用, 包括最小化重写外部闪存的操作。

- 能量有效性。WSNs 重编程应该尽可能减少对能量的消耗, 避免某些节点快速失效, 已达到延长 WSNs 生命周期的目的。

上述 6 个 WSNs 重编程的性能指标既是 WSNs 重编程的标准, 又是 WSNs 重编程设计优化的目标。此外, 有些指标之间存在着相互的矛盾需要在设计 WSNs 重编程服务时, 针对特定的应用上下文环境对它们进行权衡和最终的优化。

## 4 无线传感器网络重编程研究进展

本节我们以第 2 节介绍的重编程概念模型中与 WSNs 直接相关的代码优化、代码分发协议和节点执行环境 3 个方面来阐述 WSNs 重编程技术当前的研究进展。

### 4.1 代码优化

代码优化的目的是最小化更新代码的大小, 这对 WSNs 重编程的性能提高起着重要的作用, 它会直接影响代码传输的能量消耗, 进而影响 WSNs 的生命周期。

#### 4.1.1 直接压缩和解压缩方法

通常是利用现有的压缩算法来减小更新的代码量。尽管该方法易于实现, 但在解压缩算法的选择上需要仔细地权衡。因为解压缩后的代码对存储容量的需求大于 RAM 的容量, 这使得解压缩过程不得不对外部闪存 EEPROM 进行重写操作, 可能最终导致在计算密集的解压缩期间所消耗的能量比传输压缩的更新代码时节省的能量更多<sup>[10]</sup>。

#### 4.1.2 增量补丁方法

当新的程序版本与旧的程序版本的代码相差不大时(如只是一些参数的调整和 Bug 的修改), 只将新旧版本之间的增量作为更新补丁传输到所需的目标节点中。Reijers 和 Langendoen 使用一个 diff 脚本来描述新旧版本之间的增量, 该脚本被传送到目标节点后, 通过一个 patch list 命令完成程序地址的移动, 使新程序映像获得重构<sup>[16]</sup>。Jeong 和 Culler 借鉴 Rsync 算法<sup>[17]</sup> 解决了 Reijers 方法中存在的平台相关问题, 实现了一个与具体硬件平台无关的增量补丁方法<sup>[18]</sup>。但是以上两种方法都只是比较了代码字节, 没有考虑应用结构的特点。

#### 4.1.3 动态重定位方法

是通过提供一个具有动态加载模块能力的节点操作系统<sup>[14, 15]</sup>, 实现代码更新模块的动态重定位。这种方法由于提高了节点执行环境的能力, 使 WSNs 重编程不用传输完整的程序映像, 而只需传输独立的代码模块, 从而起到减少更新代价的作用。但是节点操作系统内核对存储容量提出了更高的要求。

#### 4.1.4 虚拟机脚本方法

采用一种完全不同的代码优化方案<sup>[19]</sup>, 该方法提供一种更高层次的指令集, 即虚拟机指令, 来实现 WSNs 重编程。由于传输虚拟机脚本的开销远小于传输程序代码的开销, 它可以取得良好的代码优化效果。但是该方法也存在着一定的问题, 如在失去编程灵活性的同时, 执行虚拟机中间代码的开销也会随着程序执行次数的增多而提高。

需要指出的是以上几种方法并不是相互独立的, 它们之间可以混合运用于特定的代码优化方案中。

### 4.2 代码分发协议

通过无线链路以多跳方式分发更新的程序代码是 WSNs 重编程技术必须要解决的关键问题。它通常被当作一种特殊的数据分发问题来考虑, 目前的研究主要集中在: 如何在不可靠的 WSNs 环境中提供一种可靠、能量有效和轻量的泛洪机制上。

Trickle<sup>[20]</sup> 是一种支持虚拟机脚本持续分发的维护服务。它通过周期性地广播代码的元数据(meta-data) 来检测是否需要代码更新。Trickle 通过借鉴 SRM(Scalable Reliable Multicast)<sup>[21]</sup> 中的抑制机制, 采用一个称为“礼貌闲聊”(polite gossip) 的方法, 改进 SPIN-RL<sup>[22]</sup> 的 ADV-REQ-DATA 三次握手机制来减少由广播风暴引起的数据冗余, 从而显著地降低能耗, 延长 WSNs 网络的生命周期。MOAP<sup>[23]</sup> 是支持完整代码映像以多跳方式进行数据传输的代码分发协议。它使用订阅-发布机制来减少不必要的消息接收和发送节点的数量, 同时采用滑动窗口方法来确保代码的可靠传输。Deluge<sup>[24]</sup> 是另一种与 MOAP 类似的 WSNs 重编程代码分发协议。它扩展了 Trickle, 支持更有效的大数据量代码分发。Deluge 将

大段数据分成固定大小的页,以页为单位通过“流水线”(Pipelining)技术来提高 WSNs 的并行传输能力。MNP<sup>[25]</sup> 针对 Deluge 和 MOAP 对重编程 WSNs 时所有节点都必须处于活动状态所造成的不必要空闲侦听问题进行改进,从而在尽可能减小传输延迟的情况下,通过关闭通信模块来节省节点能量,实验数据表明 MNP 可以用比 Deluge 多 30% 的时间传输几乎近 3 倍的数据;同时 MNP 采用贪婪算法进行发送节点的选择来解决隐藏终端引起的包冲突问题。Infuse<sup>[26]</sup> 是一种基于 TDMA MAC 的可靠代码分发协议,由于采用了基于调度的 MAC 协议,使 Infuse 避免了由广播风暴引起的广播冗余、竞争和冲突。另外,它改进了滑动窗口的 GBN 和 GBN-PP 算法,提出了基于选择的重传恢复机制保证数据的可靠传输。Naik 等人提出的 Splinkler<sup>[27]</sup> 也是一个基于 TD-

MA 的 WSNs 可靠代码分发服务。它通过计算最小连通支配集(Minimum connected dominating Set,简称 MCDS)方法来进一步优化发送节点的数目,提高能量使用的有效性和 TDMA 调度的性能。Firecracker<sup>[28]</sup> 是用来提高在 WSNs 上传输虚拟机代码速度的数据分发协议。它采用了两阶段代码分发策略,在第一阶段使用独立的路由机制将更新代码传输到少数目标节点之中,之后在第二阶段利用局部的广播机制来完成最终的重编程任务,Firecracker 试图结合路由和广播这两种数据传输方式的优势来加快代码分发的速度。

此外,随着对 WSNs 重编程技术研究的不断深入,还有一些代码分发协议将“按需”的代码更新策略与有效的泛洪机制结合起来考虑,如 TinyCubus<sup>[29]</sup>, Aqueduct<sup>[30]</sup> 等。我们通过表 1 对以上所介绍的代码分发协议进行总结和比较。

表 1 WSNs 重编程技术中代码分发协议的分类比较

Name	Size of Code	MAC	Scope	Hierarchy	Pipelining	Heterogeneity
Trickle <sup>[20]</sup>	VM script	CSMA	every node	×	×	×
MOAP <sup>[23]</sup>	Entire code image	CSMA	every node	×	×	×
Deluge <sup>[24]</sup>	Entire code image	CSMA	every node	×	√	×
MNP <sup>[25]</sup>	Entire code image	CSMA	every node	×	√	×
INFUSE <sup>[26]</sup>	Entire code image	TDMA	every node	√	√	×
Splinkler <sup>[27]</sup>	Entire code image	TDMA	every node	√	×	×
Firecracker <sup>[28]</sup>	VM script	CSMA	every node	√	×	×
TinyCubus <sup>[29]</sup>	Modular update	CSMA	Selected nodes	×	×	√
Aqueduct <sup>[30]</sup>	Entire code image	CSMA	Selected nodes	×	√	√

### 4.3 节点执行环境

WSNs 所能提供的节点执行环境直接影响着程序更新的粒度和更新时代码分发的代价,它是 WSNs 重编程的基本构件。由于 WSNs 资源极端受限的特点使得 WSNs 中除少量的网关和簇头节点外,大部分节点硬件都不具有内存管理单元(Memory Management Unit,简称 MMU),导致 WSNs 的软件更新不像传统网络那样容易实现。目前对节点执行环境的研究主要可以分成 4 类:一体化执行环境、模块化执行环境、虚拟机执行环境和中间件执行环境。

#### 4.3.1 一体化执行环境(Monolithic Environment) 如 TinyOS<sup>[13]</sup> 和 MOS<sup>[31]</sup>

它们通过在编译时静态优化代码的方法,达到对 CPU 和内存资源的合理使用,提高系统运行效率的目的。由于该方法是将操作系统和应用程序混合编译成一个单一的可执行映像,因此在一体化执行环境中,不管是应用程序还是操作系统在更新时,传输和更新代价都会很大,这给 WSNs 重编程带来了巨大的挑战,正如前面所介绍的如何减小更新代码的大小,如何在单一地址空间里提供程序间的保护以及如何确保运行环境的安全等问题,都需要进一步的研究和解决。

#### 4.3.2 模块化执行环境(Modular Environment) 如 SOS<sup>[14]</sup> 和 Contiki<sup>[15]</sup>

它们将执行环境分成内核和可动态加载模块两部分,支持简单的最匹配固定块内存分配策略。由于模块之间是一种松散的耦合,因此模块之间的调用比简单的函数调用需要更大的开销,同时由于缺乏编译时的全局优化,因此在代码的执行效率上,比一体化执行环境要低。但是模块化执行环境具有动态的模块加载能力,使得在 WSNs 重编程时,只需更新部分的程序模块,而不用将整个程序的映像全部更新。模块化执行环境在传输和更新代价上比一体化执行环境要小,因此能节省更多的能量,延长 WSNs 的生命周期。同时从 WSNs 重编程的难易程度来说,模块化执行环境相对一体化执行环境要容易。

#### 4.3.3 虚拟机执行环境(Virtual Machine Environment) 如 Mate<sup>[19]</sup> 和 ASVM<sup>[32]</sup>

它们是在操作系统之上提供一个更高级的执行环境。在进行 WSNs 重编时,由于传输的是虚拟机脚本,其代码量远低于程序原先的二进制映像代码量,因此在虚拟机执行环境中,更新代码在网络中的传输开销比前两种执行环境都要小,同时虚拟机解释执行的特点,使程序在执行时不能直接对硬件进行访问,而只能通过虚拟机以间接的方式完成,因此虚拟机执行环境具有良好的代码安全性。但是中间代码解释执行的效率比前两种执行环境都低。另外,如何选定适合的虚拟机指令集是一个挑战性的问题。ASVM 提供了一种静态的应用相关的虚拟机生成框架,它可以针对应用的特点来定义相应的虚拟机执行环境。DAViM<sup>[5]</sup> 和 DVM<sup>[33]</sup> 都试图提供一种支持可动态重配置的通用虚拟机执行环境来进一步提高 WSNs 重编程的能力。

#### 4.3.4 中间件执行环境(Middleware Environment)

是一种不同的 WSNs 重编程执行环境。由于 WSNs 表现出越来越明显的异构性特征,借鉴传统的中间件方法来提供一种适用于 WSNs 的编程和部署机制,已成为 WSNs 研究的一个热点<sup>[34]</sup>,如 Agilla<sup>[35]</sup> 是一个基于移动 Agent 机制的部署中间件。它通过注入不同的 Agent 来快速地完成应用部署实现 WSNs 的编程。

**结束语** WSNs 是一种全新的信息获取和处理技术,有着巨大的应用前景。WSNs 重编程技术是 WSNs 能否实用化的关键技术。目前 WSNs 重编程技术仍有大量的问题需要解决,比如:如何实现“按需”的代码分发;如何在异构的 WSNs 上实现重编程;如何实现有效的代码验证和获取;如何确保 WSNs 重编程过程的安全性问题等等。在本文中,我们介绍了 WSNs 重编程的概念和基本组件,给出了 WSNs 重编程面临的挑战和评价标准,从不同的方面总结和归纳了 WSNs 重编程技术近年来已有的研究,并对一些未来可能的

(下转第 112 页)

- [24] Dave Plonka's FlowScan. <http://net.doit.wisc.edu/~plonka/FlowScan/>
- [25] Detecting Worms and Abnormal Activities with NetFlow. <http://www.anheng.com/news/22/588.html>
- [26] Yegneswaran V, Barford P, Ullrich J. Internet Intrusions: Global characteristics and Prevalence. In ACM SIGMETRICS, San Diego, June 2003
- [27] Markopoulou A, Iannaccone G, Bhattacharyya S, et al. Characterization of Failures in an IP Backbone. In IEEE INFOCOM, Hong Kong, April 2004
- [28] Weaver N, Paxson V, Staniford S, et al. A Taxonomy of Computer Worms ACM CCS Workshop on Rapid Malcode (WORM), October 2003
- [29] Hussain A, Heidemann J, Papadopoulos C. A Framework for Classifying Denial of Service Attacks. In ACM SIGCOMM, Karlsruhe, August 2003
- [30] Ye N. A markov chain model of temporal behavior for anomaly detection// Workshop on Information Assurance and Security, West Point, NY, June 2000
- [31] Paxson V. BRO: A System for Detecting Network Intruders in Real Time//Proceedings of the 7<sup>th</sup> USENIX Security Symposium, 1998
- [32] Roesch M. The SNORT Network Intrusion Detection System. <http://www.snort.org>, 2002
- [33] Moore D, Voelker G, Savage S. Inferring Internet Denial-of-Service Activity// Proceedings of the 10th USENIX Security Symposium, 2001
- [34] 邹柏贤. 一种网络异常实时检测方法. 计算机学报, 2003, 26(8): 940-947
- [35] 任勋益, 王汝传, 王海艳. 基于自相似检测 DDoS 攻击的小波分析方法. 通信学报, 2006, 27(5)
- [36] 孙钦东, 张德运, 高鹏. 基于时间序列分析的分布式拒绝服务攻击检测[J]. 计算机学报, 2005, 28(5): 768-773

(上接第 68 页)

研究方向进行了简述。

### 参考文献

- [1] Campbell S, Eisenman N, Lane, E, et al. People-Centric Urban Sensing (Invited Paper)// Proc. of IEEE/ICST/CREATENET 2nd Int'l Conf. on Wireless Internet, Boston, Aug. 2006
- [2] 马华东, 陶丹. 多媒体传感器网络及其研究进展. 软件学报, 2006, 17(9): 2013-2028
- [3] Li, S Lin Y, Son S H, et al. Event detection services using data service middleware in distributed sensor networks. Telecommunication Systems, 2004, 26(2-4): 351-368
- [4] Han S, Rengaswamy R, Shea R S, et al. Sensor Network Software Update Management: A Survey. International Journal of Network Management, July 2005
- [5] Michiels S, Horre W, Joosen W, et al. DAViM: a Dynamically Adaptable Virtual Machine for Sensor Networks// Proceedings of the Workshop on Middleware for Sensor Networks, Melbourne, Australia, November 2006
- [6] Wang Qiang, Zhu Yaoyao, Cheng Liang. Reprogramming Wireless Sensor Networks: Challenges and Approaches. In IEEE Network, May 2006
- [7] IBM Tivoli Configuration Manager. <http://www-306.ibm.com/software/tivoli/products/configmgr/>
- [8] HP OpenView Change and Configuration Management. <http://www.managementsoftware.hp.com/solutions/ascm/index.html>
- [9] Microsoft Systems Management Server. <http://www.microsoft.com/smsserver>
- [10] Barr K, Asanovic K. Energy aware lossless data compression// The First International Conference on Mobile Systems, Applications, and Services, San Francisco, CA, May 2003
- [11] Estrin D, Govindan R, Heidemann J, et al. Next century challenges: scalable coordination in sensor networks. ACM MobiCom'99, Washington, USA, 1999: 263-270
- [12] Ni S Y, Tseng Y C, Chen Y S, et al. The broadcast storm problem in a mobile ad hoc network//MobiCom'99; Proceedings of the 5th Annual ACM/IEEE international Conference on Mobile Computing and Networking. ACM Press, 1999: 151-162
- [13] Levis P, Gay D, Handziski V, et al. Adam Wolisz. T2: A Second Generation OS For Embedded Sensor Networks. Technical Report TKN-05-007. Telecommunication Networks Group, Technische Universität Berlin, 2005
- [14] Han C C, Kumar E, Shea E, et al. A dynamic operating system for sensor nodes. Technical Report NESL-TM-2004-11-01. University of California Los Angeles. Networked Embedded Systems Lab, Nov. 2004
- [15] Dunkels A, Schmidt O, Voigt T, et al. Protothreads: Simplifying Event-Driven Programming of Memory-Constrained Embedded Systems//Proc. of 4th ACM SenSys'06. Boulder, Colorado, USA, Nov. 2006
- [16] Reijers N, Langendoen K. Efficient code distribution in wireless sensor networks//Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications. San Diego, CA, USA, 2003: 60-67
- [17] Tridgell A. Efficient Algorithms for Sorting and Synchronization. PhD thesis. Australian National University, 1999
- [18] Jeong J, Culler D. Incremental network programming for wireless sensors //Proceedings of the First IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks IEEE SECON. 2004
- [19] Levis P, Culler D. Mate: A tiny virtual machine for sensor networks// International Conference on Architectural Support for Programming Languages and Operating Systems, San Jose, CA, USA, Oct. 2002
- [20] Levis P, Patel P, Culler D, et al. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks//Proceedings of the First Symposium on Networked Systems Design and Implementation. San Francisco, California, United States, 2004: 15-28
- [21] Floyd S, Jacobson V, Liu C G, et al. A reliable multicast framework for light-weight sessions and application level framing. IEEE/ACM Transactions on Networking, 1997, 5(6): 784-803
- [22] Heinzelman W R, Kulik J, Balakrishnan H. Adaptive protocols for information dissemination in wireless sensor networks//MobiCom '99; Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking. ACM Press, 1999: 174-185
- [23] Stathopoulos T, Heidemann J, Estrin D. A Remote Code Update Mechanism for Wireless Sensor Networks. CENS Tech. Report # 30. Centre for Embedded Networked Sensing, UCLA, 2003
- [24] Hui J, Culler D. The Dynamic Behavior of a Data Dissemination Protocol for Network Reprogramming at Scale//Proc. of 2nd ACM SenSys'04. Baltimore, Maryland, USA, 2004: 81-94
- [25] Kulkarni S S, Wang L. MNP: Multihop Network Reprogramming Service for Sensor Networks//Proc. of IEEE ICDCS 05. 2005: 7-16
- [26] Kulkarni S S, Arumugam M. Infuse: A TDMA Based Data Dissemination Protocol for Sensor Networks. Technical Report MSU-CSE-04-46. Dept. of Computer Science and Engineering, Michigan State University, MI, 2004
- [27] Naik V, et al. Sprinkler: A Reliable and Energy Efficient Data Dissemination Service for Wireless Embedded Devices//26th IEEE Real-Time Sys. Symp. Dec. 2005
- [28] Levis P, Culler D. The Firecracker Protocol//Proc. of 11th ACM SIGOPS Euro, Wksp. Leuven, Belgium, Sept. 2004
- [29] Marrón P J, et al. Management and Configuration Issues for Sensor Networks. Int'l. J. Network Mgmt, 2005, 15(4): 235-253
- [30] Phillips L A. Aqueduct: Robust and Efficient Code Propagation in Heterogeneous Wireless Sensor Networks. Master's thesis. Univ. CO, 2005
- [31] Abrach H, Bhatti S, Carlson J, et al. Mantis: system support for multimodal networks of in-situ sensors//Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications. San Diego, CA, USA, 2003: 50-59
- [32] Levis P, Gay D, Culler D. Active sensor networks//Proceedings of the Second Symposium on Networked Systems Design and Implementation (NSDI2005). San Francisco, CA, USA, Mar. 2005: 29-42
- [33] Balani R, Han C C, Rengaswamy R K, et al. Multi-level software reconfiguration for sensor networks//ACM Conference on Embedded Systems Software (EMSOFT). Seoul, Korea, October 2006
- [34] Henriksen K, Robinson R. Middleware for Sensor Network: State-of-the-Art and Future Directions. ACM MidSens06, Melbourne, Australia, November -December 2006
- [35] Fok CL, Roman G-C, Lu C. Mobile Agent Middleware for Sensor Networks: An Application Case Study//Proc. 4th Intl. Symposium on Information Processing in Sensor Networks (IPSN'05). UCLA, Los Angeles, California, USA, April 2005: 382-387