

# 基于嗅探的无线传感器网络溯源机制的研究与设计

张亚平, 金暄宏, 尹雨欣

上海理工大学光电信息与计算机工程学院, 上海

收稿日期: 2022年9月7日; 录用日期: 2022年9月23日; 发布日期: 2022年10月8日

## 摘要

为了减小无线传感器网络(Wireless Sensor Network, WSN)溯源中节点的能量消耗, 本文通过部署嗅探节点的方式进行溯源。随着网络规模的扩大, 这种方式与传统溯源方式相比, 将大大减少能量的消耗。首先, 实现了基于CTP(Collection Tree Protocol)协议的多跳网络。然后, 在多跳网络的基础上安排嗅探节点, 并对封装信息进行一定的更改。每个节点封装自己的ID并发送。当网络中增加节点时, 嗅探节点就能够嗅探到该ID信息。通过串口发送到PC机, 通过分析PC机上串口助手接收到的数据就能够得到路径信息。实验结果表明, 通过部署嗅探节点的方式溯源, 能够大大减少能量的消耗。随着网络规模的增大, 这种方式的优点会更加明显。

## 关键词

无线传感器网络, 溯源, CTP协议, 嗅探节点

# Research and Design of Source Tracing Mechanism in Wireless Sensor Networks Based on Sniffing

Yaping Zhang, Xuanhong Jin, Yuxin Yin

School of Optoelectronic Information and Computer Engineering,  
University of Shanghai for Science and Technology, Shanghai

Received: Sep. 7<sup>th</sup>, 2022; accepted: Sep. 23<sup>rd</sup>, 2022; published: Oct. 8<sup>th</sup>, 2022

## Abstract

In order to reduce the energy consumption of nodes in Wireless Sensor Network (WSN) source

tracing, this paper deploys sniffing nodes for source tracing. With the expansion of the network scale, this method will greatly reduce energy consumption compared with the traditional source tracing method. First, a multi-hop network based on the Collection Tree Protocol (CTP) protocol is implemented. Then, the sniffing nodes are arranged on the basis of the multi-hop network, and the encapsulated information is changed to some extent. Each node encapsulates its own ID and sends it. When a node is added to the network, the sniffing node can sniff the ID information, and then send it to the PC through the serial port. The path information can be obtained by analyzing the data received by the serial port assistant on the PC. The experimental results show that energy consumption can be greatly reduced by deploying sniffer nodes to trace the source. With the increase in network size, the advantages of this method will become more obvious.

## Keywords

Wireless Sensor Network, Traceability, CTP Protocol, Sniff Nodes

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

随着科学技术的发展, 物联网逐渐发展起来, 而无线传感器网络作为物联网的关键技术, 具有广阔的应用前景和巨大的商业应用价值。无线传感器网络是一种分布式传感网络[1], 已在多个领域如智能交通[2]、环境监测[3]等领域得到应用。在有的无线传感器网络中, 节点的能量由电池提供, 系统需要根据用户的需要, 由较多的传感器节点相互合作才能完成, 这些传感器节点之间必定会进行数据传输, 这些传输将会消耗较多的能量。无线传感器网络具有动态、低速、低能耗、硬件资源受限、无人值守等特点, 通过对参与节点进行可信度评估并进行后续处理, 可以显著提高网络的鲁棒性和可靠性, 而评估可信度的前提就是对 WSN 中数据包的溯源路径进行重建。传统的方法是基于 WSN 节点自身, 通过分组路径中各节点将自身 NODE ID 叠加到所转发的分组中, 最后在基站(Base Station, BS)进行解码, 最终恢复出原始路径。然而, 在分组转发过程中消耗的能量远远大于数据处理过程中消耗的能量, 因此, 随着路径的增长, 传统溯源方式会导致转发的分组中数据过载, 从而过度消耗节点能量, 这对于资源受限的 WSN 环境是不能接受的, 因此, 如何减少能量的消耗对于无线传感网的设计尤为重要。

近年来, 针对 WSN 中的溯源问题, 各国学者纷纷展开研究, 并且都提出了一些解决方案。由于受到节点能量、传输带宽、存储空间以及环境变化等因素的影响, 不同方法的性能表现出很大的差异。毛健[4]等人提出了一种基于生成树的溯源数据压缩方法, 其基本思想是在字典中存放 WSN 拓扑图的生成树并对其建立索引, 在数据包传输过程中传输的是生成树的索引而不是完整的生成树。在大规模稀疏 WSN 中采用该方法, 溯源数据在文件大小和传输能耗等方面都优于已知的其它溯源数据编码技术。徐芹宝[5]等人提出一种基于路径索引差分的溯源编码方法, 进一步提高了溯源的平均压缩比, 不仅可以有效克服已知的溯源编码方法对网络拓扑结构变化敏感的问题, 而且可以实现更高的溯源平均压缩比。Sy Denh 等提出了一种叫做 CAPTRA (Coordin Ated Packet TR Aceback)的方法[6], CAPTRA 利用节点发送分组信息时的广播特性, 将分组信息分散存储在附近的监听节点中。当需要取回信息时, 转发节点和监听节点交换信息, 最终就能构造出整条路径。

为了尽可能降低节点的能量消耗,本文提出了基于嗅探进行溯源的方式,在基于CTP协议[7]的多跳网络中部署嗅探节点,根据嗅探节点接收到的信息得到路径信息,从而实现溯源,有效减少了节点中能量的消耗。

## 2. CTP 协议

CTP是一种基于树状结构的汇聚协议[8],网络中的一些节点设为根节点,其余节点根据路由梯度选择父节点作为下一跳,隐式地选择根节点,最终形成到根节点的路由。基于TinyOS的CTP协议总体框架,如图1所示,包含三部分:链路估计器、路由引擎和转发引擎。位于底层的链路估计器用于估计节点与邻居节点之间的链路质量并维护邻居表,CTP中使用ETX(Expected number of Transmission)来表示链路质量的估计值,ETX越小,则链路质量越好。路由引擎则根据链路估计器的信息选择父节点并维护路由表路由引擎的操作,有两个主要任务和一个主要事件:updateRouteTask被定期调用并选择一个新的父对象,sendBeaconTask将当前路由信息广播到邻居,主要事件是接收邻居的信标,该信标将更新邻居表。链路估计器和路由引擎之间可以通过接口LinkEstimator和CompareBit交流,其中,LinkEstimator.getQuality()的返回值越小,则通信链路质量越好。转发引擎、维护转发队列并选择队头的包转发给父节点,并在发生错误时进行重传。转发引擎对其传输包进行计时。它区分四种传输情况:转发、成功、确认失败和环路检测。在每种情况下,转发引擎在发送下一个数据包之前都要等待一段随机的时间。其目标是防止相关流量(例如,路由转发数据包中的节点)干扰自身。

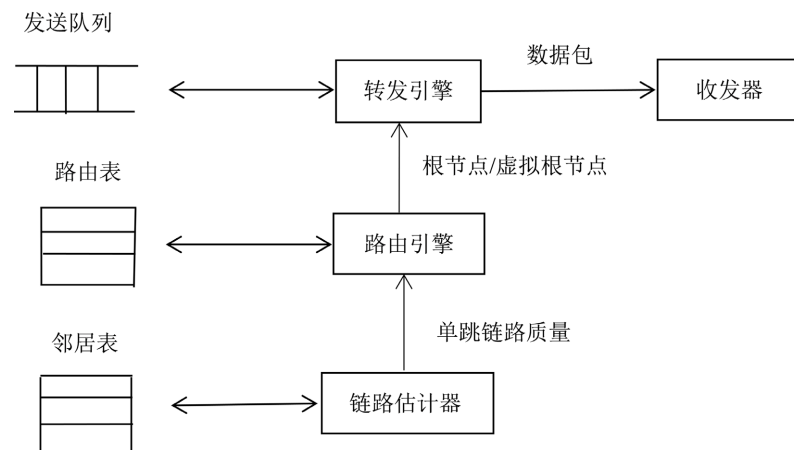


Figure 1. Overall architecture of CTP

图 1. CTP 总体架构

## 3. 系统设计

### 3.1. 系统软件设计

#### 3.1.1. 基于CTP协议实现多跳转发

本次设计中多跳网络部署是基于CTP协议的,因此,首先需要在“multihopsend”文件夹下的“Makefile”中编写代码:

```
COMPONENT=multihopsendAppC
CFLAGS += -I$(TOSDIR)/lib/net/ -I$(TOSDIR)/lib/net/ctp -I$(TOSDIR)/lib/net/4bitle
include $(MAKERULES)
```

然后分别在“multihopSendAppC.nc”和“multihopSendC.nc”中添加组件和接口，最后在“multihopSendC.nc”中添加实现多跳转发的主组件。基于CTP的多跳转发实现流程可用图2表示，调用RadioControl.startDone()后进行判断，若节点的ID为1，则调用RootControl.setRoot()将该节点设置为根节点，只能由根节点从Receive接口接收消息；若节点ID不是1，说明不是根节点，则需要继续转发。

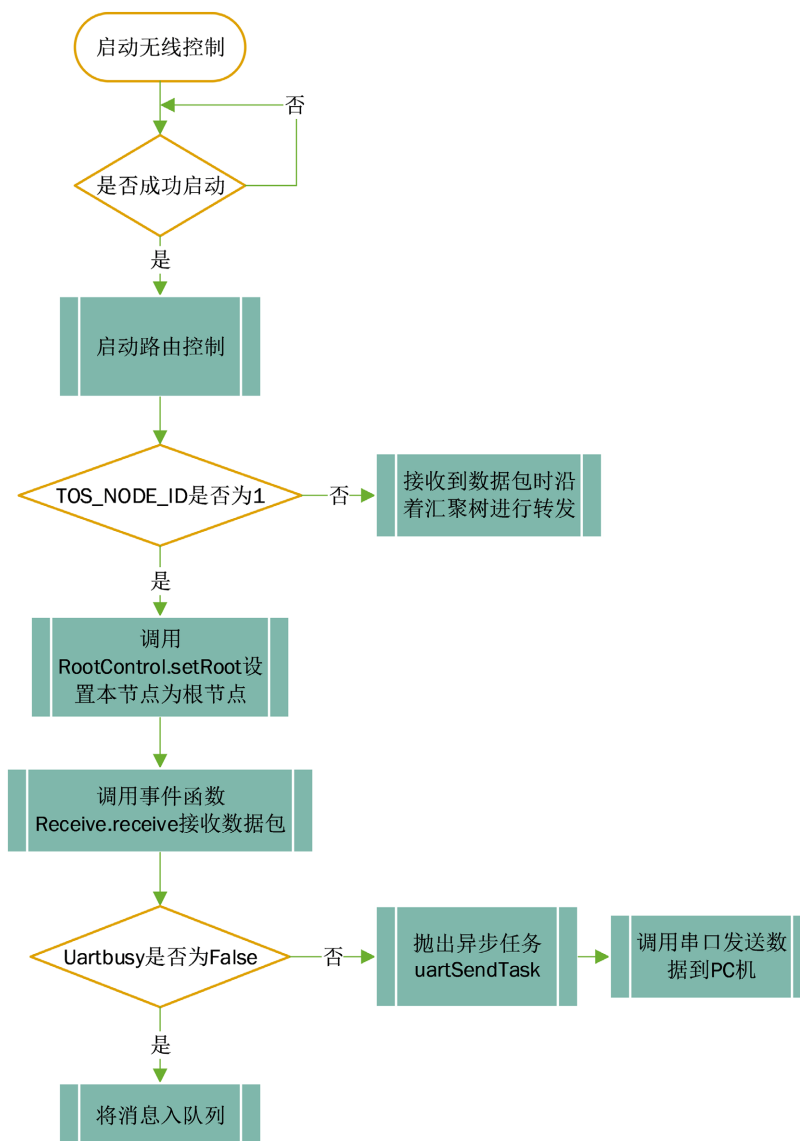


Figure 2. Multi-hop forwarding flow chart  
图2. 多跳转发流程图

### 3.1.2. 实现嗅探功能

实现嗅探功能的程序位于TinyOS2.1/apps/BaseStation中，BaseStation是充当简单活动消息的应用程序串行和无线电链路之间的桥梁。它取代了GenericBaseTinyOS1.0和TinyOS1.1的TOSBase。在串行连接上，BaseStation发送和接收简单活动消息(不是特定的无线电数据包)；在无线电链路上，它发送无线电活动消息。BaseStation会将其已编译的组ID复制到消息中从串行连接移动到无线电，并会过滤掉传入的不包含该组ID的无线电消息。BaseStation包括双向队列，并确保一条消息进入队列后，它将最终留在另

一条消息上。队列使 BaseStation 可以更多地处理负载峰值。仅当该消息成功入队以传递到无线链路后，BaseStation 才确认该消息通过串行链路到达。首先，需要在“Makefile”中进行预设置：CFLAGS += -DCC2420\_NO\_ACKNOWLEDGEMENTS; CFLAGS += -DCC2420\_NO\_ADDRESS\_RECOGNITION; 即允许基站嗅探来自任何发射机的数据包并防止基站错误地确认数据包。然后在对应文件中添加实现嗅探功能并通过串口转发的主组件，实现流程可用图 3 表示。

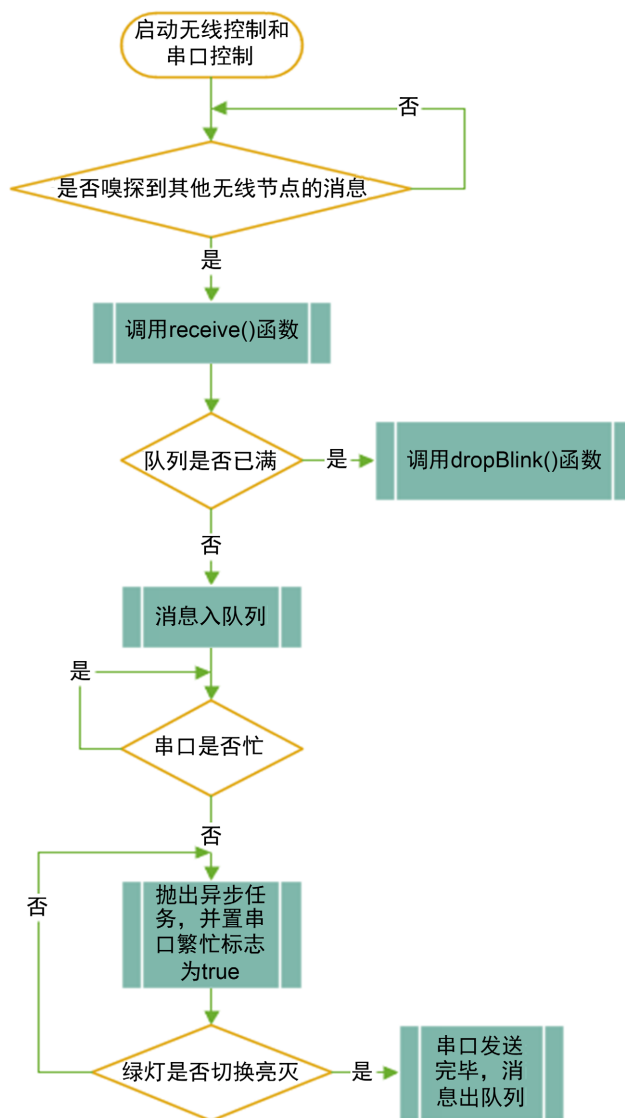


Figure 3. Flow chart of sniffing implementation

图 3. 嗅探实现流程图

### 3.2. 系统硬件设计

首先，进行基于 CTP 协议的多跳转发部署：将汇聚节点与 mib520 编程器相连后插入 PC 机，打开各节点的开关，打开串口助手接收数据。

然后，进行嗅探节点设计，由于本次设计中是要实现消息从无线电桥接到串行，因此，需要将嗅探节点与 mib520 编程器连接后插入 PC 机，打开各节点的开关，打开串口助手接收数据。

将编程器通过 USB 插入 PC 机时，由于 mib520 编程器能够将 USB 虚拟为串口，因此，PC 机成功识别到该 USB 接口后，应该能够在设备管理器的端口和通用串行总线控制器中分别看到 USB Serial Port (COM4 和 COM5)、USB Serial Converter A 和 B。

## 4. 实验结果

### 4.1. 实验设置

如图 4 所示，利用指令下载对应程序到节点，其中 Micaz 为本次设计所用的节点，mib510 为编程板，本次设计所用的编程板为 mib520，这里写 mib510 即可，因为 mib520 引用了 mib510 的设置，如下：

```
#$Id: mib520.extra,v 1.1 2008-04-23 20:31:58 sallai Exp $
# The MIB520 uses the same protocol as the MIB510, so we
# just include mib510.extra
MIB510 = $(MIB520)
include $(TINYOS_MAKE_PATH)/avr/mib510.extra
```

在 ubuntu 中 dev 文件夹下存储了设备文件，包括计算机的所有外部设备，其中 dev/ttyUSB0 为 USB 转串口终端，接 USB 转串口线可以使用该端口设备。

```
root@zyp-virtual-machine:/home/zyp/tinyos/tinyos-release-tinyos-2_1_2/apps/Ba
tation# make micaz install,2 mib510,/dev/ttyUSB0
mkdir -p build/micaz
```

Figure 4. Schematic diagram of code download

图 4. 代码下载示意图

本次设计中使用的是 mib520 编程器，提供的是 USB 接口，该编程板能够实现将 USB 接口虚拟为串口。下载程序时会提示无法找到 dev/ttyUSB0，因此需要进行 USB 转串口操作，转换步骤如下。

首先，利用 `sudo apt-get install minicom` 命令安装 minicom，并利用 `sudo apt-get install lrzsz` 命令安装 lrzsz 软件包。安装完成后配置 minicom。

如图 5 所示，运行 `sudo minicom-s`，然后，选择 Serial port setup。

```
+-----[configuration]-----+
|  Filenames and paths          |
|  File transfer protocols      |
|  Serial port setup           |
|  Modem and dialing           |
|  Screen and keyboard         |
|  Save setup as dfl           |
|  Save setup as..            |
|  Exit                        |
|  Exit from Minicom          |
+-----+-----+

```

Figure 5. Select serial port settings

图 5. 选择串口设置

如图 6 所示，选择 A 将 Serial Device 修改为：/dev/ttyUSB0；保存后退出。

如图 7 所示，首先，进入 minicom 菜单，然后，进入配置文件所在目录，选择“Filenames and paths”，然后分别选择 A, B 修改 Download directory 和 Upload directory 为：mib510 -->PC 和 PC -->mib510。

```

+-----+
| A - Serial Device      : /dev/ttyUSB0
| B - Lockfile Location  : /var/lock
| C - Callin Program    :
| D - Callout Program   :
| E - Bps/Par/Bits      : 115200 8N1
| F - Hardware Flow Control : No
| G - Software Flow Control : No
|
| Change which setting?
+-----+

```

Figure 6. Set serial port device  
图 6. 设置串口设备

```

+-----[configuration]-----+
| Filenames and paths
| File transfer protocols
| Serial port setup
| Modem and dialing
| Screen and keyboard
| Save setup as dfl
| Save setup as..
| Exit
+-----+
Comp+-----+
Port| A - Download directory : mib510 -->PC
| B - Upload directory   : PC -->mib510
Pres| C - Script directory  :
| D - Script program     : runscript
| E - Kermit program     :
| F - Logging options
|
| Change which setting? |
+-----+

```

Figure 7. USB and serial port conversion  
图 7. USB 与串口转换

## 4.2. 实验结果和分析

如图 8 所示，当嗅探节点接收到数据并成功调用串口发送向 PC 机发送数据后，绿灯会切换亮灭。

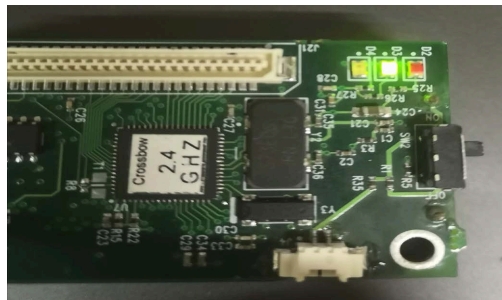


Figure 8. Node successfully sniffed data  
图 8. 节点成功嗅探到数据

同时，在串口助手选择 COM5 串口，波特率为 57600，选择 hex 显示、加时间戳和分包显示。

串口助手接收到的嗅探数据如图 9 所示，0x(93)为组件 SerialAMSenderC 的参数，这里同样结点 msg->ID 的类型定义为 nx\_uint16\_t；嗅探节点能够嗅探每个节点的数据包，路径中每增加一个节点，嗅探节点能够嗅探到该节点发送的 ID 号，并将嗅探到的数据入队列后通过串口发送收到 PC 机，通过观察即可得出路径信息。其中，ID 为 0x22 的节点为发送数据包节点，ID 为 0x03 的节点为转发节点，ID 为 0x02 的节点为嗅探节点。图 9 中红色框中数据即为嗅探到的路径信息。

```

|[21:26:11.583]收←◆7E 45 00 00 22 00 03 0A 22 71 00 00 01 A7 00 03 35 93 00 03 4B A2 7E

```

Figure 9. Sniffing path information  
图 9. 嗅探到路径信息

## 5. 结束语

本次设计首先实现了基于 CTP 协议的多跳转发, 为了方便观察收发的数据, 在汇聚节点上增加转发数据包到 PC 机的程序; 在此基础上实现了嗅探节点, 可嗅探到其他节点的 ID 号, 并通过串口发送到 PC 机, 通过 PC 机上的串口助手进行接收后就可以观察数据包转发路径。这样一来, 网络中每增加一个节点, 嗅探节点都可以嗅探到对应的 ID, 而不需要每增加一个节点就需要将 ID 封装到数据包中。随着网络规模的增加, 通过嗅探方式实现溯源的方法能够充分发挥节省能量消耗的作用。

## 参考文献

- [1] 唐菁敏, 曲文博, 苏慧慧, 郑锦文. 一种基于帝企鹅差分算法的 WSN 覆盖优化[J]. 云南大学学报(自然科学版), 2021, 43(1): 46-51.
- [2] Anisi, M.H. and Abdullah, A.H. (2016) Efficient Data Reporting in Intelligent Transportation Systems. *Networks Spatial Economics*, **16**, 623-642. <https://doi.org/10.1007/s11067-015-9291-9>
- [3] Gao, R., Zhou, H. and Su, G. (2011) A Wireless Sensor Network Environment Monitoring System Based on TinyOS. In: Yin, Z.X., Ed., *Proceedings of 2011 International Conference on Electronics and Optoelectronics*, Dalian, 29-31 July 2011, V1-497-V1-501.
- [4] 毛健, 王昌达. 基于生成树的 WSN 溯源数据压缩方法[J]. 软件导刊, 2017, 16(7): 14-17.
- [5] 徐芹宝. WSN 中基于路径索引差分的溯源数据压缩方法[D]: [硕士学位论文]. 镇江: 江苏大学, 2018.
- [6] Cuha, S., Rastogi, R. and Shim, K. (1998) CURE: An Efficient Clustering Algorithm for Large Databases. In: Hass, L.M. and Tiwary, A., Eds., *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, New York, June 1998, 73-84. <https://doi.org/10.1145/276305.276312>
- [7] Gnawali, O., Fonseca, R., Jamieson, K., Moss, D., & Levis, P. (2009) Collection Tree Protocol. *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, Berkeley, 4-6 November 2009, 1-14. <https://doi.org/10.1145/1644038.1644040>
- [8] 李晨, 施伟斌. CTP 协议的原理及性能测试[J]. 装备制造技术, 2012(2): 88-90+93.