

# 无线传感器网络的 RPL 路由协议研究

朱琳, 高德云, 罗洪斌

(北京交通大学 电子信息工程学院 北京 100044)

**摘要:** 路由协议执行网络拓扑描述、路由选择和数据包转发功能, 影响整个网络的性能和存活时间。现有的路由协议需要发送大量数据包维护网络拓扑, 以及大量的存储空间来存储路由条目。由于硬件的限制, 无线传感器节点无论是能量, 还是其处理能力、存储能力都受到极大的制约。因此, IETF RoLL 工作组提出了一种针对低功耗有损网络的 IPv6 路由协议, 即 RPL 路由协议。文中对 RPL 路由协议的拓扑构建过程、数据包路由过程和 Trickle 定时器的算法等进行了分析, 通过使用 COOJA 仿真工具对其进行仿真, 验证了 RPL 路由协议在低功耗有损网络中具有较高的性能。

**关键词:** 无线传感器网络; IPv6; RPL 路由协议

中图分类号: TP31

文献标识码: A

文章编号: 1673-629X(2012)08-0001-04

## Research of RPL Protocol of Wireless Sensor Network

ZHU Lin, GAO De-yun, LUO Hong-bin

(School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China)

**Abstract:** Routing protocol implements the description of network topology, routing path selection and data packets forwarding, which influences the performance and alive time of the entire network. Existing routing protocols require a large amount of messages to maintain network topology, as well as large storage space to keep routing entries. Due to the constraints of hardware, the energy and the ability of processing, also the storing of wireless sensor network nodes has been dramatically limited. Hence, the IETF routing over low-power and lossy networks (RoLL) working group has recently proposed the IPv6 routing protocol for low power and lossy networks, i. e., the RPL protocol. In this paper, the topology building process and data packets routing process, also the Trickle timer algorithm of RPL routing protocol is analyzed. It has been simulated by COOJA, which clearly shows that RPL routing protocol plays a beneficial and rewarding role in low-power and lossy networks.

**Key words:** wireless sensor network; IPv6; RPL routing protocol

## 0 引言

无线传感器网络是由一系列体积小、造价低的节点组成, 在监测区域内部或附近自组织地构成网络<sup>[1~3]</sup>。这些节点普遍具有传输速率低、处理能力和能量储备有限等问题。另外, 对于网络整体而言, 通信的可靠程度、链路失效问题, 都要求网络具有相当成熟的技术以建立和保持可靠有效的路径<sup>[4, 5]</sup>。IETF RoLL 工作组评估过目前已经存在的路由协议, 如 OSPF, IS-IS, AODV, OLSR, 由于低功耗有损网络 (Low-Power and Lossy Networks, LLN) 的上述特点, 导致目前已存在的路由协议并不适合 LLN。因此该工作组研究制定了 RPL (Routing Protocol for LLN) 协

议<sup>[6~8]</sup>。RPL 协议是一个距离矢量路由协议, 节点通过交换距离矢量构造一个有向无环图 (Directed Acyclic Graph, DAG)。DAG 可以有效防止路由环路问题, 其根节点通过广播方式与其余节点交互信息, 然后节点通过路由度量来选择最优的路径<sup>[9, 10]</sup>。

文中详细分析了 RPL 路由协议的拓扑构建过程、数据包路由过程和 Trickle 定时器算法等, 同时使用 COOJA 工具<sup>[11~13]</sup>对 RPL 路由协议进行了仿真, 验证其性能。

## 1 RPL: Routing Protocol for LLN

RPL 是为 LLN 而设计的距离矢量路由协议, 通过使用目标函数 (Object Function, OF) 和度量集合构建具有目的地的有向无环图 (Destination Oriented Directed Acyclic Graph, DODAG)。目标函数利用度量和约束条件的集合计算出最优路径。由于网络部署的目的性有异, 同一个网络可能需要不同的链路质量要求等等, 导致在相同的网络中有可能有几个目标函数。

收稿日期: 2012-01-09; 修回日期: 2012-04-13

基金项目: 国家自然科学基金 (60802016, 60972010)

作者简介: 朱琳 (1988-), 女, 硕士研究生, 研究方向为无线传感器网络; 罗洪斌, 博士, 教授, 研究方向为网络体系结构、通信网络技术。

### 1.1 DODAG 的构建过程

RPL 路由协议规定了一系列的新的 ICMPv6 控制消息,以交换图的相关信息和构建拓扑。它们分别是:请求信息(DODAG Information Solicitation, DIS), DODAG 信息对象(DODAG Information Object, DIO),目的地通告(Destination Advertisement Object, DAO)。

■ 图的构建过程从根或边界路由器(LoWPAN Border Router, LBR)开始。根首先使用 DIO 消息来广播有关图的信息。监听根节点的邻居节点收到并处理 DIO 消息,根据目标函数、DAG 特点、广播路径开销等等来决定是否加入到这个图当中。一旦节点加入到图当中,它就有一条通向 DODAG 根的路由,根则成为这个节点的父节点。节点接着会计算自己在图中的 rank 值,并向自己的父节点发送包含路由前缀信息的 DAO 消息。节点也可以使用 DIS 消息来主动向邻居节点来请求图信息。所有邻居节点重复这一过程,直至在整个网络中构建出一个以 LBR 为根节点的 DODAG。DODAG 的构建过程的简化流程图如图 1 所示。LBR 发送包含图相关信息的 DIO 消息,监听 LBR 的节点 A 收到之后,加入到图中,并回复包含自己前缀信息的 DAO 消息给 LBR;节点 A 发送 DIO 消息,其功率发送范围内的 B 接收后加入图中并回复 DAO 消息;网络启动后,节点 B 曾收到节点 C 发来的 DIS 消息,在节点 B 加入图之后,对节点 C 发送 DIO,邀请其加入 DODAG,节点 C 向其父节点 B 发送 DAO 消息,节点 B 收到后会整合信息,向其父节点发送 DAO 消息,逐级整合并回送 DAO 消息直到 LBR,在 LBR 处包含所有节点的前缀信息。

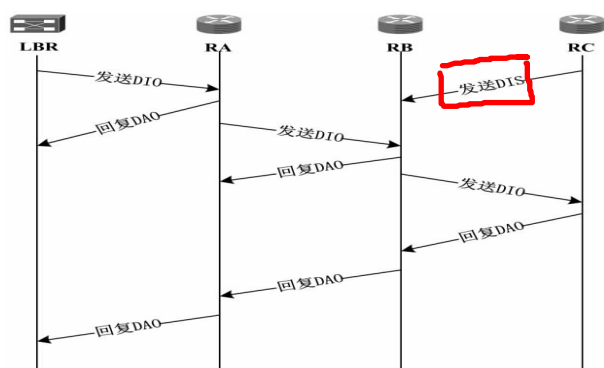


图 1 DODAG 的构建简化流程图

### 1.2 RPL 的路由过程

节点根据收到的 DIO 消息,选择父节点,添加路由,广播图信息。通过这种方式构建的图中每个节点都有通向自己的父节点的路由表项,这样叶子节点可以通过向自己父节点发送数据,一直发送至根节点,这种模型称为 MP2P(MultiPoint - To - Point),或是“向上”路由。

同样的,来自 LLN 网络外部的信息,从根节点或

是某个中间节点,目的是某个叶子节点,这种模型称为 P2MP(Point - To - MultiPoint),需要“向下”路由来实现。通常使用 DAO 消息来完成从上至下的路由建立过程。每个节点在加入图时,会给它的父节点集合发送 DAO 消息。其父节点在接收了 DAO 消息后,处理其前缀信息,并在路由表中加上一条路由条目。然后有选择性地聚合从不同节点收到的前缀信息,向它的父节点集合发送 DAO 消息。这一过程持续进行直到前缀信息到达根节点,建立出到某个前缀的完整路径。不过需要注意的是这种模式称为“存储”模式,要求所有的节点都有可用的存储空间来存储路由表。RPL 也支持“非存储”模式,即所有的中间节点不存储任何的路由信息。RPL 还支持一种 P2P 的通信模式,即在图中从某节点到其它节点的通信。

总之,节点需要向上路由时,只需传送给自己的父节点,父节点继续将之发送给自己的父节点,一直向上传送到根。然而,节点需要向下路由,或是需要执行点到点通信时,则根据网络中是“存储”或是“非存储”模式而有所不同。在“存储”模式下,节点存储子图内收到的所有 DAO 消息的前缀信息,所以 P2MP 只需根据节点存储的子图内的前缀信息,选择下一跳节点即可, P2P 路由只需向上传送到与目的节点的共同祖先,再从祖先节点处选择下一跳节点发送直至目的节点。而在“非存储”模式下,节点存储空间非常有限,所有的前缀信息都存储在 LBR 处。节点向自己的父节点汇报 DAO 消息,在图的根节点处汇总整合所有的前缀信息,从而形成向下路由。P2MP 或是 P2P 路由都需要将数据包传送到根节点处,这样当根节点收到了一个目的地为图中某个节点的数据包后,它会将数据包的源路由头部 option - field 加上路径上的所有节点前缀信息,然后将其发送到下一跳子节点。路径上的每一个中间节点检查源路由头部,再发送给下一跳节点。反复这一过程直至目的节点<sup>[8-9]</sup>。

### 1.3 环路避免和环路检测

在传统网络中,由于拓扑改变和节点间未及时同步的问题,可能会导致临时性的环路产生。为了减少数据包的丢失、链路拥塞的情况,必须尽快检测出环路。在 LLN 中,环路的影响是有限的,并且这种环路的产生可能是暂时的,所以过度反应反而会导致更大程度上的路由碰撞和能量消耗。因此, RPL 的策略是不保证不会出现环路,而是试图避免环路的出现。RPL 定义了两种规则来避免环路,这两种规则都依赖于节点的“rank”值。

1、最大深度规则,不允许节点选择图中深度更大(rank 值更大)的节点作为自己的父节点。

2、拒绝节点“贪婪”规则,不允许节点试图移动到

图中更深的地方,以增加自己潜在父节点的数量。

RPL 的环路检测策略是在 RPL 的路由头部中设置相关的 bit 位,通过这些 bit 位来检测数据的有效性。比如,当一个节点将数据包发向自己的一个子节点,将 bit 位置成“down”,然后将数据包发送到下一跳节点。收到“down”bit 的数据包的节点,查询自己的路由表,发现数据包是在“向上”方向传输的,则证明出现了环路,此时数据包需要被丢弃,并触发本地修复。

#### 1.4 全局修复和本地修复

RPL 支持在链路或节点失效之后的修复机制。RPL 支持两种修复机制:全局修复和本地修复。当检测到链路或邻居节点失效后,节点在向上方向上没有其他路由器,则本地修复被快速触发以寻找替代父节点或路径。当本地修复发生时,有可能破坏了整个网络的最优模式,从而由根节点触发全局修复机制重建 DODAG,图中的每个节点都重新运行目标函数来重新选择更优的父节点。

#### 1.5 Trickle 定时器管理

RPL 中的定时器与其他运行于受限条件少的环境中的路由协议不同。大多数的路由协议使用周期性的“keepalive”来保持邻居关系、更新路由表。但在 LLN 中,资源极度受限,所以显然这种方式是不合适的。RPL 使用了一种称为“Trickle Timer”的定时器来完成自适应的定时器机制,它可以控制发送 DIO 消息的速率。算法将构建图的过程视为一个一致性问题(当节点检测到网络中出现环路,或是节点加入或移出网络都视为不一致现象),使用 Trickle 计时器来决定何时广播 DIO 消息。当网络趋于稳定时,RPL 消息的数量减少;检测到不一致现象时,重置定时器加快发送 DIO 消息以便快速解决问题。

Trickle 算法可描述如下:假设  $I_{\min}$  为最小时间间隔,  $I_{\max}$  为最大时间间隔,  $I$  为当前时间间隔,  $t$  为当前时间点,  $K$  为冗余常量,  $C$  为计数器,  $I_{\text{double}}$  为  $I$  能够自乘 2 的最大次数。

(1) 开始。设置  $I$  为  $[I_{\min}, I_{\max}]$  中的一个值。开始第一个时隙。

(2) 第一个时隙开始,置  $C = 0$ ,  $t = [I/2, I)$  中的一个随机点。时隙终止在  $I$  处。

(3) Trickle 收到一个一致性传输,就让  $C = C + 1$ 。

(4) 在时间  $t$ , Trickle 检查是否有  $C < K$ , 且只有在  $C < K$  时允许发送数据包。

(5) 当  $I$  到期,就使  $I = I * 2$ , 如  $I * 2 > I_{\max}$ , 置  $I = I_{\max}$ 。

(6) 如果 Trickle 接收到不一致性传输, Trickle 为响应外部 events 重置 timer。

Trickle 算法只有在第 4 步才传输,表示在检测到不一致性并做出反应之间有一个固定的时间间隔。立

即反应有可能会引发广播风暴。

## 2 协议仿真及分析

文中对 RPL 路由协议的实现过程进行仿真及分析,使用 COOJA 作为仿真工具进行仿真实验。设置仿真的感知区域为一块边长为  $L = 200\text{m}$  的方形区域。在环境中只有一个 LBR 节点,9 个普通节点;所有节点的功率范围均为  $R_{\text{tx}} = 75\text{m}$ ;所有节点在其功率发送范围内的丢包率均为 0;设置节点以随机方式分布在感知区域内;节点的平台选用 sky。产生 DIO 消息的时间间隔,初始值为  $1\text{s}$ ,然后该时间间隔随着仿真时间而变化。设  $I_{\min} = 1\text{s}$ ,  $I_{\text{double}} = 16$ ,所以在两个连续 DIO 消息的最大时间间隔为  $18.2$  小时<sup>[14,15]</sup>。网络中的其他节点会发送 DAO 消息。这是为了实现向下方向的 P2P 和 P2MP,节点收到从子节点发送来的 DAO 消息,从而建立子节点前缀信息的路由表。在本仿真中,假定每个节点都有能力存储网络中其他节点的路由信息,因此不需要源路由。

为了说明 RPL 路由所形成拓扑的过程,也就是 DODAG 形成的过程,对所有节点形成的拓扑图进行分析,如图 2 所示。图中节点附近显示的是根据目标函数计算得的 rank 值,节点根据 rank 值选择自己的父节点,从而构建此 DODAG。在仿真中,待网络稳定后,将节点 5 由图示白色位置移动到当前位置,如图有,  $\text{rank}(1) = 1.0$ ,  $\text{rank}(10) = 2.0$ ,  $\text{rank}(6) = 3.0$ ,  $\text{rank}(8) = 4.0$ ,  $\text{rank}(5) = 8.5$ 。根节点的 rank 值为 1,其他节点距离根节点的跳数越大,其 rank 值越大。在节点 5 移动到新位置,网络稳定后,对所有节点的接收包数量、丢包数量、距离 LBR 节点的跳数、综合计算得出的 Routing Metrics 以及根据 Routing Metrics 得出的量化值 ETX 进行统计,如表 1 所示。根据节点的信息与网络拓扑容易发现网络拓扑的形成是与节点的 Routing-Metric 和 ETX 相关的。

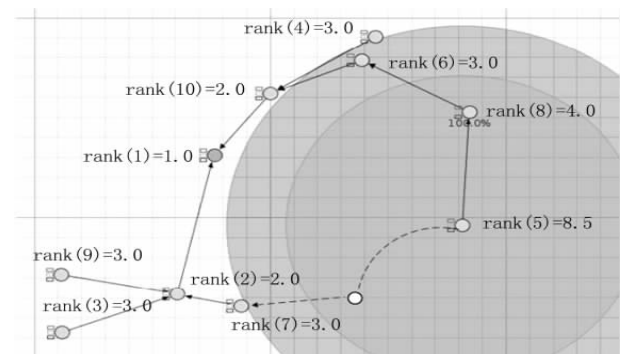


图2 节点使用 RPL 构建 DODAG

RPL 路由协议在网络中实现的过程,即 LBR 节点启动过程、节点收到 DIO 消息并扩散 DIO 消息、节点发送 DIS 请求消息以及节点向父节点发送 DAO 消息



表 1 节点的相关信息汇总表

Node	Received	Lost	Hops	Rmetric	BTX	LDC	TDC
1.1	0	0	0.000	0.000	0.000	0.000	0.000
2.2	41	0	1.000	514.341	1.000	0.700	0.107
3.3	41	0	2.000	810.537	1.152	0.620	0.124
4.4	41	0	2.000	805.463	1.107	0.643	0.107
5.5	33	8	3.333	1373.091	1.932	0.617	0.262
6.6	41	0	2.000	789.073	1.058	0.671	0.140
7.7	41	0	2.000	779.317	1.034	0.643	0.107
8.8	41	0	3.000	1084.878	1.146	0.641	0.135
9.9	41	0	2.000	801.561	1.119	0.637	0.119
10.1	41	0	1.000	518.634	1.018	0.689	0.110
Avg	40.111	0.889	2.037	830.766	1.174	0.651	0.135

建立路由。在仿真中,将上述四个过程的部分 ICMP 消息截取出来,如图 3 所示。LBR 启动后,广播包含图信息的 DIO 消息。在监听 LBR 的节点收到 DIO 后会计算自己的 rank,将 LBR 加入到自己的邻居缓存中,如图 3 第一段所示。也有一些节点主动发送 DIS 进行请求,接收到 DIS 请求的节点会回复 DIO,如图 3 第二段所示。加入 DAG 的节点继续发送 DIO 消息,监听到此 DIO 消息的节点受邀加入网络,如图 3 第三段所示。节点加入网络后会向父节点发送 DAO 消息,其父节点收到 DAO 消息后向根节点发送,根节点收到后整合前缀信息,如图 3 第四段所示。

↑	ID:1	SEND PREFIX IN DIO FOR AAAA::
↑	ID:1	SEND MULTICAST DIO WITH RANK 256
↓	ID:2	RECEIVE DIO FROM FE80::212:7401:1:101
↓	ID:2	NEIGHBOR FE80::212:7401:1:101 ADD TO CACHE
↓	ID:2	INCOMING DIO RANK 256
↑	ID:4	SEND A DIS
↓	ID:10	RECEIVE A DIS FROM FE80::212:7404:4:404
↓	ID:10	MULTICAST DIS => RESET DIO TIMER
↑	ID:10	SEND PREFIX IN DIO FOR AAAA::
↑	ID:10	SEND MULTICAST DIO WITH RANK 1024
↓	ID:6	RECEIVE DIO FROM FE80::212:740A:A:A0A
↓	ID:6	NEIGHBOR FE80::212:740A:A:A0A ADD TO CACHE
↑	ID:10	RECEIVE DAO FROM FE80::212:7404:4:404
↑	ID:10	DAO PREFIX: AAAA::212:7404:4:404
↑	ID:10	FORWARD DAO TO PARENT FE80::212:7401:1:101
↓	ID:1	RECEIVE DAO FROM FE80::212:740A:A:A0A
↓	ID:1	DAO PREFIX: AAAA::212:7404:4:404
↑	ID:10	SEND PREFIX IN DIO FOR AAAA::
↑	ID:10	SEND MULTICAST DIO FOR AAAA::
↓	ID:1	RECEIVE A DIO FROM FE80::212:740A:A:A0A
↓	ID:1	NEIGHBOR ALREADY IN CACHE
↓	ID:1	INCOMING DIO RANK 1088
↓	ID:1	DIO CONF: DBL=8 MIN=12 RED=10 MAXINC=768
↓	ID:1	COPY PREFIX INFO

图 3 节点启用 RPL 路由协议后的 ICMP 消息

为验证 Trickle 定时器在 RPL 路由中能够有效地调节节点发送数据包的速率,在发生不一致现象时可以快速反应,特别对所有节点的发送 DIO 的间隔时间进行监控,如图 4 所示。此时间间隔由 Trickle 定时器控制,初始  $I_{\min} = 1s$ ,每次  $I$  到期后,就令  $I = 2 * I$ 。随着网络的稳定,其发送 DIO 消息间隔就越大,节点发送的消息数越少。由于仿真中人为地将节点 5 移动,

导致图中节点 5 的 Beacon Interval 由 2200s 突然降至 1s。这是由于节点 5 的移动,被视为一个“不一致现象”产生,从而节点 5 触发了本地修复,按 Trickle 算法将  $I$  置于  $I_{\min}$ ,待其稳定后,仍会按“阶梯”式增长。图中易看出节点先后到达一个阶梯上,这主要是由于 Trickle 算法在第 4 步需要检测当前积累的一致性传输的数量是否小于冗余常量,只有在满足这一条件时才会发送数据包,因而造成了由发现一个不一致现象到立即反应之间有一个固定时延,这也是为了应对所有节点在检测到不一致现象后同时反应会造成广播风暴。因而不同节点的状态变化之间也存在一个固定时延。图 5 所示为节点 5 在移动后,由于大量的发送 DIS 信息,与其他节点频繁进行信息交互,从而导致节点 5 的功耗急剧上升。

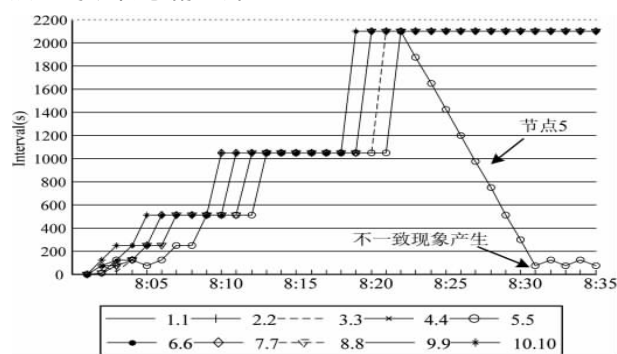


图 4 节点的 DIO 发送间隔变化

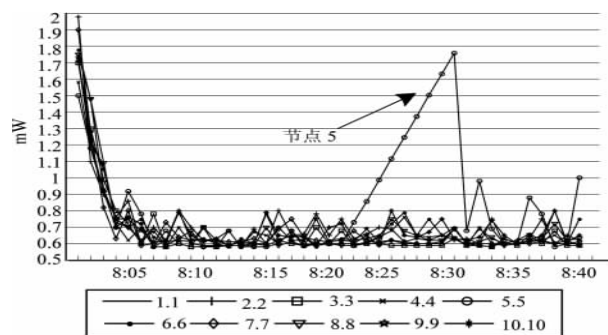


图 5 节点的功耗随时间变化情况

### 3 结束语

RPL 路由协议是由 RoLL 工作组提出,目的是解决低功耗有损网络的路由问题。RPL 路由协议使用 ICMP 的消息格式加以改进,形成三种消息 DIO、DIS 和 DAO 来实现路由过程。其目标函数并未明确要求,使得不同网络可以根据不同的功能需求来确定不同的目标函数,从而构建合适的网络拓扑。文中利用 COOJA 的仿真结果可以容易得到,利用 Trickle 定时器能有效地控制发送信息的数量,随着网络稳定,数据包的数量大大减少;利用 RPL 路由协议可以快速检测到网络突发事件并做出反应。在未来的研究中,将着眼于 RPL

(下转第 8 页)

图 3 中第一行到第三行,分别为噪声密度为 0.2, 0.5 和 0.8 的噪声图像以及在不同滤波算法下处理的结果。

## 5 结束语

文中提出了一种在各种噪声密度下高效、快速地去除椒盐噪声的滤波算法。该算法把图像中的噪声点和非噪声点进行分类,只针对噪声点进行滤波。根据窗口中的噪声点数量来自适应改变窗口的大小,并在去噪处理过程中采用改进的非对称裁剪中值滤波来计算中值。实验结果表明图像不但在滤波性能客观评价指标上明显优于传统中值滤波及其改进算法,而且该算法计算复杂度简单,高效快速。

### 参考文献:

- [1] Gonzalez R C, Woods R E. Digital image processing [M]. 2nd ed. New Jersey: Prentice - Hall 2001.
- [2] Tukey J W. Exploratory data analysis [M]. Reading, MA: Addison - Wesley, 1971: 98 - 224.
- [3] Sun Tong, Neuvo Y. Detail - preserving Median - based Filters in Image Processing [J]. Pattern Recognition Letters, 1994, 15(4): 341 - 347.
- [4] Zhou Wang, David Z. Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted

Images [J]. IEEE Transactions on Circuits and Systems: Analog and Digital Signal Processing, 1999, 46(1): 78 - 80.

- [5] Hwang H, Haddad R A. Adaptive median filters: new algorithms and results [J]. IEEE Trans on Signal Processing, 1995, 4(4): 499 - 502.
- [6] Eng H L, Ma K K. Noise adaptive soft - switching median filter [J]. IEEE Trans. on Image Process, 2001, 10(2): 242 - 251.
- [7] 杨恒伏, 陈孝威. 小波域鲁棒自适应公开水印技术 [J]. 软件学报, 2003, 14(9): 1652 - 1660.
- [8] 刑藏菊, 王守觉, 邓浩江, 等. 一种基于极值中值的滤波算法 [J]. 中国图象图形学报, 2001, 6(6): 533 - 536.
- [9] 曲延锋, 徐键, 李卫军, 等. 有效去除图像中脉冲噪声的新型滤波算法 [J]. 计算机辅助设计与图形学报, 2003, 15(4): 397 - 401.
- [10] 金良海, 熊才权, 李德华. 自适应型中心加权的中值滤波器 [J]. 华中科技大学学报, 2008, 36(8): 9 - 12.
- [11] 陈初侠, 丁勇, 刘栋莉. 去除椒盐噪声的自适应开关加权均值滤波 [J]. 计算机工程, 2010, 36(4): 210 - 212.
- [12] 常瑞娜, 穆晓敏, 杨守义, 等. 基于中值的自适应均值滤波算法 [J]. 计算机工程与设计, 2008, 29(16): 4257 - 4259.
- [13] Srinivasan K S, Ebenezer D. A New Fast and Efficient Decision - based Algorithm for Removal of High - density Impulse Noises [J]. IEEE Signal Processing Letters, 2007, 14(3): 189 - 192.

(上接第 4 页)

路由协议的本地修复机制,致力于实现节点不需要等待新的 DIO 消息或是更高的 DAG 序列号,就能寻找到一个新的父节点<sup>[16, 17]</sup>。

### 参考文献:

- [1] 朱燕, 黄明科. 物联网的发展和应探讨 [J]. 电信网技术, 2010(11): 34 - 37.
- [2] 孙利民, 李建中. 无线传感器网络 [M]. 北京: 清华大学出版社, 2005.
- [3] 张宏科, 梁露露, 高德云. IPv6 无线传感器网络的研究及其应用 [J]. 中兴通信技术, 2009, 15(5): 37 - 40.
- [4] 李振强. IPv6 技术解密 [M]. 北京: 人民邮电出版社, 2006.
- [5] 蒋挺, 赵成林. 紫蜂技术及应用 [M]. 北京: 北京邮电大学出版社, 2006: 46 - 115.
- [6] Oliveira L M L. Routing and mobility approaches in IPv6 over LoWPAN mesh networks [J]. International Journal of Communication Systems, 2011, 24(11): 1445 - 1466.
- [7] Akkaya K. A survey of routing protocols in wireless sensor networks [J]. Elsevier Ad Hoc Network Journal, 2005, 33: 325 - 349.
- [8] Vasseur J P. RPL: the IP routing protocol designed for low power and lossy networks [S]. [s. l.]: Internet Protocol for Smart Objects (IPSO) Alliance, 2011.

- [9] 宋非, 侯乐青. 浅析智能物件网络中的 RPL 路由技术 [J]. 网络技术, 2011, 9(9): 23 - 26.
- [10] Winter T. Rpl: Routing protocol for low power and lossy networks [S]. [s. l.]: Internet Engineering Task Force, 2011.
- [11] Österlind F. An introduction to COOJA [EB/OL]. [2010 - 11]. [http://www.sics.se/contiki/wiki/index.php/An\\_Introduction\\_to\\_Cooja](http://www.sics.se/contiki/wiki/index.php/An_Introduction_to_Cooja).
- [12] Österlind F. Crosslevel sensor network simulation with COOJA [C]//Proc. of LCN. [s. l.]: [s. n.], 2006: 641 - 648.
- [13] Österlind F. A Sensor Network Simulator for the Contiki OS [R/OL]. [2006 - 05]. <http://soda.swedish-ict.se/2296/1/SICS-T-2006-05-SE.pdf>.
- [14] Levis P. The Trickle Algorithm [S]. [s. l.]: Internet Engineering Task Force, Request for Comments 6206, 2011.
- [15] Watteyne T. Implementation of Gradient Routing in Wireless Sensor Networks [J]. IEEE GLOBALCOM, 2009(11): 1 - 6.
- [16] Tripathi J. A performance evaluation study of RPL: Routing protocol for low power and lossy networks [C]//Conf. on Information Sciences and Systems. [s. l.]: [s. n.], 2010.
- [17] Accettura N. Performance Analysis of the RPL Routing Protocol [C]//Proceedings of the 2011 IEEE International Conference on Mechatronics. [s. l.]: [s. n.], 2011.