

6LoWPAN 无线心电传感网络架构与 RPL 优化仿真

胡侯华¹, 黄同成², 李平¹, 曾文飞¹, 段頊¹

(1. 邵阳学院 机械与能源工程学院, 湖南 邵阳 422000;

2. 湘西南农村信息化服务湖南省重点实验室, 湖南 邵阳 422000)

摘要: 阐述了通过心电采集节点、6LoWPAN 边界路由器构建无线传感网络, 实现了无线传感器网络 (WSN) 与 IPV6 网络之间的通信; 针对 RPL 网络协议中的 Trickle 算法机制导致选择次优链路, 影响网络路由的可靠性和稳定性问题, 提出了优化方案; 开展了仿真测试和优化后的网络性能指标分析; 测试结果和分析表明, 6LoWPAN 无线传感网络不仅可以完成 WSN 与外部 IPV6 网络之间的连接承载数据收发, 并且在 Trickle 算法优化之后, 对提高网络性能有较高影响。

关键词: 6LoWPAN; 心电监护传感网络; RPL; Trickle 优化; Cooja 仿真

6LoWPAN Wireless ECG Sensor Network Architecture and RPL Optimization Simulation

HU Yuhua¹, HUANG Tongcheng², LI Ping¹, ZENG Wenfei¹, DUAN Xu¹

(1. School of Mechanical and Energy Engineering, Shaoyang University, Shaoyang 422000, China;

2. Key Laboratory of Hunan Province for Rural Information Service in Southwest Hunan, Shaoyang 422000, China)

Abstract: Describes the construction of a wireless sensor network through ECG acquisition nodes and 6LoWPAN border routers to realize the communication between the wireless sensor network (WSN) and the IPV6 network; the Trickle algorithm mechanism in the RPL network protocol leads to the selection of sub-optimal links, which affects the reliability and stability of the network routing. This paper proposed an optimization scheme. Carried out simulation test and optimized network performance index analysis. The test results and analysis show that the 6LoWPAN wireless sensor network can not only complete the connection between WSN and the external IPV6 network to carry data transmission and reception, but also has a high impact on improving network performance after the Trickle algorithm is optimized.

Keywords: 6LoWPAN; ECG monitoring sensor network; RPL; Trickle optimization; Cooja simulation

0 引言

随着物联网技术和 AI 技术等日新月异的发展, 智慧养老成为人们关注的热点。智慧养老平台和可穿戴监测设备的发展非常迅猛, 但是面世的绝大部分心电监护设备还是采用目前常见的 ZigBee、GSM、Wifi、蓝牙等无线传输协议的模块^[1]。它们要么抗干扰能力弱, 要么网络传输能力差, 要么面临网络协议转换复杂等问题, 这些无线传输模式都是承载于 IPV4 网络下。IPV6 加 WSN (WSN, wireless sensor networks) 是目前的热门研究方向^[2]。工作组在 2004 年发起了一份新的网络方案: 6LoWPAN (IPV6 over Lo WPAN), 实现了 IPV6 和 IEEE802.15.4 网络的无缝连接^[3]。本文依据现有技术资料运用 6LoWPAN 技术架构无

线传感网络, 并且在 RPL 网络协议层面上提出了优化方案, 据此方案构造的无线传感网络协议转换更简单、网络稳定可靠、构建网络拓扑更快速。

1 传感网络总体设计

1.1 功能需求

1) 心电监护无线传感网络需要具备完整的体系架构, 包括底层数据采集传感节点、传输节点。

2) 基于 6LoWPAN 实现 WSN 与 IPV6 网络互联, 6LoWPAN 边界路由器需要具备 IPV6 网络和 IPV4 网络协议的转换功能。

3) 对 RPL 网络协议层面进行优化设计, 保证无线传感网络的稳定性与可靠性, 保证时刻变化的无线传感网络建

收稿日期: 2021-02-03; 修回日期: 2021-04-26。

基金项目: 湖南省教育厅重点科研项目(20A450); 邵阳学院研究生科研创新项目(CX2020SY016)。

作者简介: 胡侯华(1995-), 男, 安徽黄山人, 硕士研究生, 主要从事智能信息处理方向的研究。

黄同成(1964-), 男, 湖南邵阳人, 教授, 硕士研究生导师, 主要从事智能信息处理与挖掘、智能控制、智慧养老方向的研究。

曾文飞(1971-), 男, 湖南邵阳人, 教授, 硕士研究生导师, 主要从事智能信息处理、移动群智感知网络、数据挖掘方向的研究。

引用格式: 胡侯华, 黄同成, 李平, 等. 6LoWPAN 无线心电传感网络架构与 RPL 优化仿真[J]. 计算机测量与控制, 2021, 29(10): 239-243, 255.

立新拓扑更为快速。

1.2 整体架构

该无线传感网络系统由心电采集模块、6LoWPAN 边界路由器（6LBR，6LoWPAN border router）两部分组成。心电采集模块中 AD8232^[4] 数据处理单元将获取的心电信号经过处理后发送给 STC12C5A 信号采集单元。处理好的心电信号通过 6LoWPAN 网络传输至 6LoWPAN 边界路由器。最后监护终端获取由边界路由器传输至 IPV6 网络的心电数据，便于低功耗、长时间对病人生理参数进行采集^[5]。传感网络整体架构如图 1 所示。

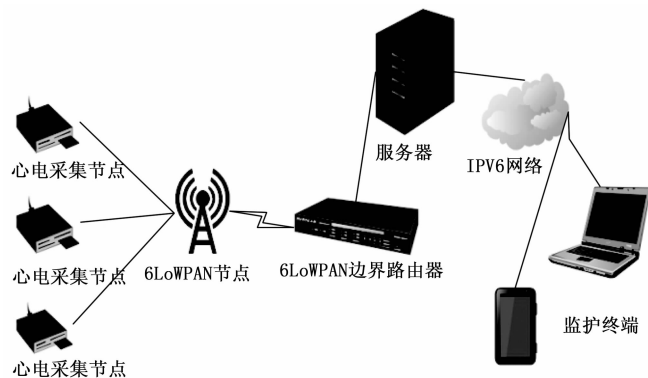


图 1 传感网络整体架构

2 传感网络硬件设计

该传感网络硬件设计包括两个部分：6LoWPAN 心电采集节点以及 6LoWPAN 边界路由器的设计。

2.1 6LoWPAN 心电采集节点设计

6LoWPAN 心电采集节点由采集模块和传输模块构成，其中采集模块包含电极、AD8232 信号处理器、STC12C5A 信号采集器。AD8232 进行高通、低通滤波处理、信号放大^[6]；STC12C5A 实现对放大器输出的信号进行模数转换，输出形式采用 UART 的方式。选择 CC2530 作为传感网络的传输模块，CC2530 使用射频技术进行数据传输，搭载 80C51 内核，配置 8KB 的 RAM 和 256KB 存储器^[7]，其多种工作模式在设计之初就考虑了低功耗要求，处理和传输心电数据的成本低。

2.2 6LoWPAN 边界路由器设计

6LoWPAN 边界路由器负责将 6LoWPAN 无线传感网络和 IPV6 网络连接起来，它要实现的功能有两个：

- 1) 作为 RPL 网络的 Root 节点，对整个 6LoWPAN 网络进行创建和控制。
- 2) 作为 6LoWPAN 无线传感网络接入 IPV6 网络的网关，将心电采集节点获取的数据传输至以太网。

边缘路由器主要由 CC2530 模块和 ENC28J60 以太网接口模块^[8]构成，6LoWPAN 边界路由器如图 2 所示。

CC2530 是应用最广泛的芯片之一，在本文中作为传输模块的核心；本文的以太网模块同样选用的是最受广泛应用的 ENC28J60 模块。

ENC28J60 模块采用串行外设接口通信（SPI），任何控

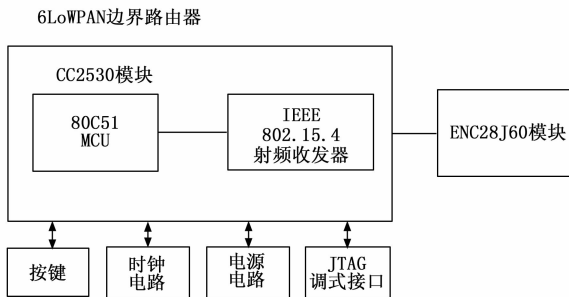


图 2 6LoWPAN 边界路由器图

制器只要带有 SPI 的以太网接口，就能与 ENC28J60 连接。ENC28J60 采用了包过滤机制来完成对传输的数据包进行限制，并且内部含有 TX/RX 缓冲器、MAC 模块、PHY 模块、DMA 模块。PHY 模块实现信号的模数转换；TX/RX 缓冲器是 8KB 的双端口 RAM，在 SPI 接口和 MAC 模块起数据缓存的作用。使用时通过两个引脚和 SPI 实现与主控制器的通信，传输速率可达 10 Mb/s^[9]。边界路由器的 6LoWPAN 节点连接 ENC28J60 模块组建成网络，侦听 6LoWPAN 和串口是否有数据。

3 软件设计与优化

在 Contiki 操作系统上开展软件设计以满足无线传感网络的协议转换及数据传输。Contiki 是为 LLN (low power and lossy networks) 开发的操作系统^[10]。开发者已经在 Contiki 中安装了 6LoWPAN 和 RPL 路由协议，提供 TCP/IP 网络协议支持，包括 IPV6，可以直接运用相关代码建立起传感网络，具有处理 WSN 数据帧和 IPV6 数据报文功能。

Cooja^[11]是一款基于 Contiki 操作系统的仿真软件^[12]，能够在 Contiki 的驱动下实现轻量级运行。工作范围很广，可以模拟大集群或小规模的几乎所有无线传感网络。而 Cooja 软件最强大方便之处在于节点执行模拟器的结果与将节点加载到固件中的结果基本相同。

3.1 6LoWPAN 心电采集节点软件设计

6LoWPAN 心电采集节点通电后，启动 Contiki 系统开始初始化，配置参数。随后侦听边界路由器的状态，若 RF 模块收到来自边界路由器发送来的数据包，进行报文重组、解压缩处理后解析此数据包。解析后的指令信息一般分为两种：监测和控制。若为前者，则采集老年人心电信号并通过 RF 发送给 6LoWPAN 边界路由器；若解析结果为后者，则执行控制指令并将心电采集模块的状态信息发送给 6LoWPAN 边界路由器。6LoWPAN 心电采集节点软件工作流程如图 3 所示。

3.2 6LoWPAN 边界路由器软件设计

6LoWPAN 边界路由器采用 CC2530+ENC28J60 实现，软件层面上使用时因 ENC28J60 的 MAC 模块遵循 IEEE802.3 标准（以太网标准），在数据传输过程中负责完成 SPI 格式和 IEEE802.3 MAC 数据帧格式^[13]之间的转换，随后需要完成自身的 MAC 地址和 IPV4 地址之间的绑定。

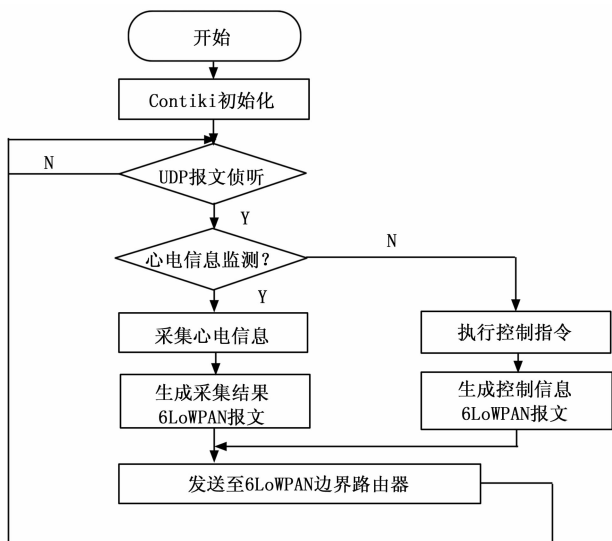


图 3 6LoWPAN 心电采集节点软件工作流程

6LoWPAN 如果是接入纯 IPV6 网络中可以直接在网络层进行数据转发, 而不需要转换协议, 但由于目前 IPV6 网络尚未大量运用, 当下多是将 6LoWPAN 网络接入 IPV4 网络中, 因此我们需要用到 NAT64 转换技术实现 6LoWPAN 数据包和 IPV4 数据包的格式转换, 并将 RPL 网络中的心电节点 IPV6 地址端口映射为 ENC28J60 以太网接口的 IPV4 地址端口, 解决 IPV4 地址不够用的问题^[14]。NAT64 协议转换模型如图 4 所示。

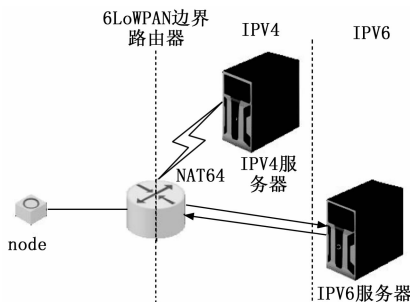


图 4 NAT64 协议转换模型图

NAT64 是一种网络地址翻译技术, 当终端节点访问远程 IPV4 服务器时, 边界路由器启动 IPV4 与 IPV6 协议转换模块 NAT64, IPV6 主机端发出封包, 源 IPV6 地址被 NAT64 模块映射成 IPV4 地址, 同时 IPV4 主机对应返回的封包, NAT64 模块遍历映射关系作出回应, 这样就完成了终端节点连接 IPV4 主机的工作。IPV6 数据报经封装后通过 IPV4 的链路层传送^[15]实现数据的相互收发。

3.3 RPL 路由及 Trickle 优化设计

3.3.1 RPL 路由协议与 Trickle 定时器

RPL (IPv6 routing protocol for low power and lossy networks) 路由协议^[16]用于智能网络, 为受限设备通过有损链路而开发^[17]。通过有向无环图 DODAG (destination oriented directed acyclic graph)^[18]创建 WSN, 根据 OF 以及节点

间相互传递信息来创建和维护。DODAG 中包含 DIO (DODAG information object) DODAG 信息对象; DIS (DODAG information solicitation) DODAG 信息请求; DAO (destination advertisement object) DODAG 目的地通告^[19]。

定时器管理机制是 RPL 协议的关键组成部分^[20]。RPL 路由协议中定时器基于 Trickle 算法来实现 DIO 控制报文管理, 这是一种低功耗有损网络的自适应机制, 适用于资源受到限制的环境。Trickle 定时器要是在架构新 DODAG 的途中查询到网络中存在闭合环路或有节点加入进来亦或是节点网络丢失等都会判定为不一致现象^[21], 当网络变得稳定时, RPL 消息数量会相应减少; 当查询到上面所提现象时, 会重置 Trickle 定时器, 加快 DIO 消息发送, 达到网络拓扑的重建目的。

Trickle 算法中含有最小时隙间隔 I_{min} (min interval)、最大时隙间隔 I_{max} (max interval)、冗余常数 K 这三个参数以及当前的时隙间隔大小 I 、当前时间值 t 、冗余事件计数器 C 三个变量。Trickle 算法简述为以下步骤: 1) 在 $[I_{min}, I_{max}]$ 中的任意位置设置一个 I 的值, 然后触发第一个时隙工作; 2) 触发第一个时隙后, 在 $[I/2, I]$ 随机选择一个点为 t , 此时 $C=0$ 。并且在 I 处停止时隙; 3) 如果传输过程中 1 次一致性传输现象被 Trickle 接收到, 那么就使 C 相应递增 1 ($C+=1$); 4) 在时间 t , Trickle 自动检测 C 的值, 并判断是否有 $C < K$, 允许发送数据包的唯一条件是有且仅有 $C < K$ 条件成立; 5) 若 I 的时效截止, 则执行 $I * 2$, 如 $I * 2 > I_{max}$, 置 $I = I_{max}$; 6) 如果出现不一致性现象时 (环路及新节点加入、脱离网络等), Trickle 为响应外部事件, 会重置定时器。

3.3.2 Trickle 算法的局限性

由于 RPL 网络在设计的时候考虑到低功耗问题出于轻量级网络思路限制了路由质量。在 DODAG 建立和维护过程中会由于路由质量原因无法体现节点拓扑“公平性”导致非最优链路的产生, 在这个问题中就牵涉到了 Trickle 算法。原始的 Trickle 算法中每个节点体现“公平性原则”即在可控条件下, 定时发送请求信息来确认路由的可靠性, 尽量使得节点在不同网络情况下保有基本相同的发送概率。但由于有损网络特性问题, 网络创建过程中时刻受到时间变化及节点加入或丢失情况影响而不停改变。

因此, 原始 Trickle 算法的规定在现实中不能真正意义上实现保有同概率的机制, 从而影响路由的可靠性和网络性能。

3.3.3 Trickle 优化设计

针对原始 Trickle 的缺陷, 我们需要针对此问题作出改进, Trickle 算法优化部分代码如下所示:

```
function Trickle-Initialization()
I = Imin, Imax = Imin * 28, k = 3 //定义 Trickle 参数及冗余常数
a = 0 //新增变量 a, 记录节点连续未发送 DIO 消息的时间间隔次数
```

```

function DIOIntervalBegins()
c = 0 //第一个时隙开始,c置0
timer = random(I/(2^(a+1)), I) //选择当前时间值 t
function ConsistentTransmissionReceived()
c = c+1 //收到一致性传输则 c 自增 1
function TimerExpires()
if (k ≥ c|a = 2) then //判断 c 与 k 的值并且若两次未成功发送 DIO 消息
Transmit DIO //发送一次 DIO 消息
a = 0 //a 置 0
else
a = a+1 //若 c<k 或者两次时间间隔内成功发送了 DIO 消息则 a 加 1
end if
function DIOIntervalEnds()
c = 0 //时隙结束,c 置 0
if InconsistentTransmissionReceived
I = Imin
a = 0 //当收到不一致传输现象则置 I=Imin
else
I = I × 2
if Imax ≤ I then
I = Imax //若 Imax 小于 I,则置 I 为 Imax
end if
end if

```

原 Trickle 算法代码中每个节点受发送时间间隔设定影响,上轮未发送节点并不一定能在本轮被选中发送 DIO 消息,因此我们需要在 Trickle 中改变为此时间间隔来获取相对公平。值得注意的是,并不需要设置其提前于其他节点发送消息,这样可以更多的减少对其他节点的影响。从而达到某个节点被抑制的次数越多,其下一轮被发送的概率就越大目的。我们设定每个节点维护一个新增变量 a , 变量 a 记作未发送 DIO 消息事件,保存各个节点未发送的时间间隔次数。在 DODAG 创建拓扑时一个节点在当前时间值 t 未能成功发送 DIO 消息, a 事件值加一;相反,则将 a 事件值清空为 0。每个节点获取每轮不同的发送优先级均通过维护自己的事件值来实现。另外区别于初始算法, a 事件值还通过更改时间间隔的每一轮监听和分配节点的持续时间来实现优先级。在上面 Trickle 优化方案里,每个时间间隔都与原算法不同。侦听和发送的两个时间段不是均匀分布的,而是相对于变量以可变间隔设置。侦听时间为 $(0, I/2^{-(a+1)})$, 发送时间为 $(I/2^{-(a+1)}, I)$ 。

我们同时规定,当某个节点连续两个时间间隔都未能成功发送 DIO 消息(因为节点在两次发送未果的情况下相比于其他节点有更高的发送优先级,考虑到有可能未发送节点较多时万一会出现此问题),则在第三个时间间隔内传送节点在某一随机时刻 t 必须要发送 DIO 消息,随后维护 a 值置为 0。后续通过仿真实验与初始 RPL 网络协议进行对比,以此来验证该优化方案的可行性。

4 RPL 路由仿真实验

本文的仿真实验基于 Ubuntu 系统,使用 InstantContiki3.0 作为开发环境,Contiki 系统中已有相应的应用程序,在网络层中我们使用的是 RPL 路由协议,因此设计此仿真实验以验证本文方案的网络通信可行性。

4.1 仿真环境设计

运行 InstantContiki3.0 系统的 Terminal 终端,输入以下命令打开 cooja 仿真器:

```

cd contiki/tools/cooja/
ant run

```

打开 Cooja 后,在 Cooja 工作界面建立 100 m × 100 m 的仿真环境,Trickle 算法的参数设置为 $I_{min}=2^8$ ms, $I_{max}=2^{(8+8)}$ ms,冗余常数 $K=3$ 。创建 21 个节点,内部运行 RPL + UDP 协议,首先创建一个节点 1 选取 sky mote 类型,并修改 Description 作为 RPL 组网协议里的 root 节点,运行 contiki/examples/ipv6/rpl-border-router/border-router.c 程序,并选取另外 20 个节点分别编号 2-21 作为感知节点,在其内部运行 contiki/examples/er-rest-example/er-example-server.c 程序。仿真环境如图 5 所示。

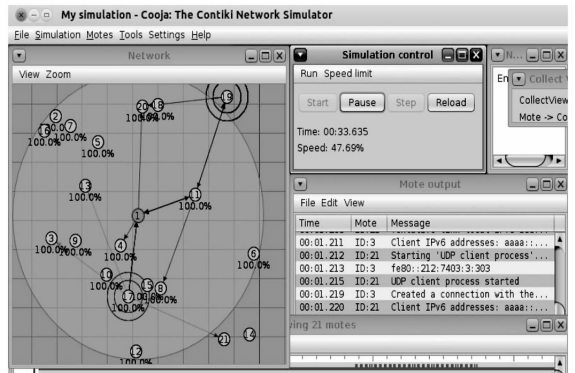


图 5 仿真环境图

节点 1 作为 6LoWPAN 边界路由器节点向外发送 DIO 消息建立 DODAG,邻居节点若在 root 节点射频范围内且侦听到 DIO 消息会选择加入到节点 1 的 DODAG。邻居节点以 root 节点 1 作为父节点,向节点 1 回复 DAO 消息表示自己已加入有向无环图,而后邻居节点向外发送 DIO 包。诸如此类,其他节点也各自寻找到父节点,并回复 DAO 包,加入 DODAG。

4.2 仿真结果验证和分析

4.2.1 验证网络连通性

在 Cooja 仿真界面可以查看到各个节点的路由信息,如图 7 所示。因此此 6LoWPAN 网络中的每个节点都是根据 ContikiMAC 地址自动生成的,因此所有的邻居节点都是以“fe08”为开头的 ipv6 地址。根据 Cooja 路由信息框中获取的 6LBR 地址,直接打开 InstantContiki3.0 自带浏览器输入地址查询即可。在 InstantContiki3.0 系统中还可以采用在 Terminal 窗口中输入 Ping6 + IPv6 地址的方式验证网络联通

性。Cooja 界面各节点路由信息如图 6 所示。

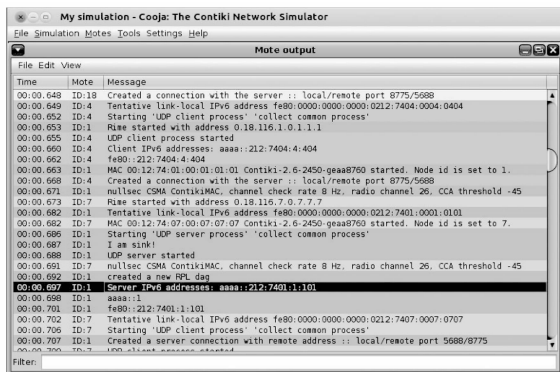


图 6 cooja 界面各节点路由信息

查看路由信息可知, 6LoWPAN 边界路由器已经根据 RPL 路由协议和邻居发现协议自动生成了 6LoWPAN 网络, 并且可以实现从外部网络访问 6LoWPAN 边界路由器节点。仿真测试结果表明, 该无线传感网络可以由 RPL 路由协议和邻居发现协议生成 6LoWPAN 网络, 实现无线传感网络 WSN 和 IPV6 网络的互联互通, 成功架构了基于 6LoWPAN 的心电监护初始无线传感网络。

4.2.2 Trickle 算法优化结果分析

为验证 Trickle 算法优化前后的 RPL 网络性能, 我们从以下两个方向进行对比: 业务报文的投递成功率、网络的平均链路质量, 如图 7 (a)、图 7 (b)、图 7 (c) 所示。

1) 业务报文投递成功率: 在一个时间区间内, 计算单位时间内目的节点正确接收到的报文数和源节点发送的总报文数比率。

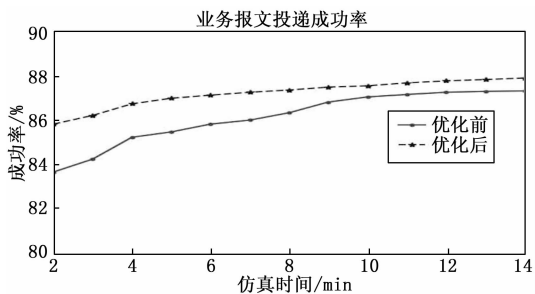
2) 节点到 6LBR 的平均链路质量: 定义为仿真环境中节点到 6LBR 路径上的平均链路度量值。

3) 节点平均能耗: 定义为仿真环境中各 node 节点随时间变化的平均能耗。

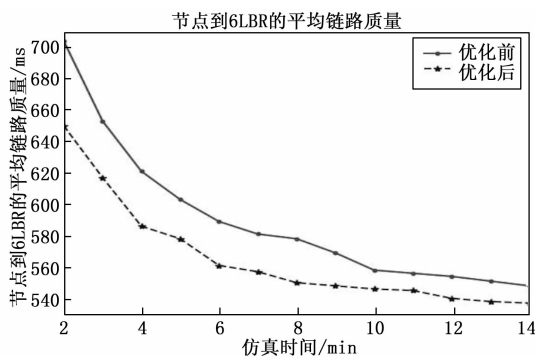
优化前后的仿真数据对比如图 7 所示, 从图中可以看到从建立 DODAG 拓扑到趋于稳定的过程中, 经过 Trickle 算法优化后的方案在业务报文投递成功率和节点到 6LBR 的平均链路质量方面性能要明显高于原始 Trickle 算法的 RPL 网络, 并且经过优化后的 RPL 网络, 节点到 6LBR 的平均能耗也比初始方案低。后随着时间增长, 网络开始逐步稳定, 两种方案的网络性能指标差距也在逐步缩小, 这是因为初始 RPL 网络协议也逐步形成了更好更稳定的网络拓扑。因此, 我们可以得出这样的结论, 在降低了节点平均能耗的基础上, Trickle 算法优化后的 RPL 网络协议在业务报文投递成功率和网络链路质量上都有了明显的提升, 这对低功耗无线传感网络的拓扑频繁变化更快形成新的 DODAG 的解决思路提供了更好的方案。

5 结束语

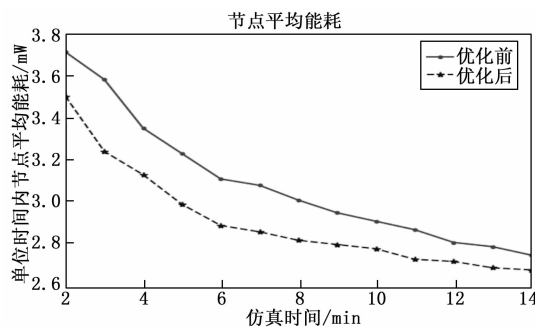
本文基于 6LoWPAN 技术设计了一种应用于心电监护系统的无线传感网络, 能够实现 WSN 监护网络与 IPV6 的



(a) 业务报文投递成功率



(b) 节点到 6LBR 的平均链路质量



(c) 节点平均能耗

图 7 优化前后对比图

的互联互通, 同时提供了一种基于 Trickle 算法优化的 RPL 改进方案, 借助仿真实验验证了该方案在提升网络拓扑性能上的作用, 能够较于原 RPL 协议更快地形成 DODAG 拓扑, 以及在降低了节点平均能耗时依然在改善无线传感网络业务报文投递成功率和提升网络链路质量上具有良好作用。该无线传感网络的设计不仅保证了网络传输的稳定性, 还提升了节点加入 DODAG 的速度, 使得组网更加快速方便, 为该无线传感系统应用提供了更优方案, 具有一定的应用前景和现实意义。

参考文献:

[1] 刘才权. 基于 6LoWPAN 的心电无线监护系统设计 [D]. 成都: 电子科技大学, 2013.
 [2] 干开峰, 王 俊, 张大敏. 6LoWPAN 无线传感网络温室监测系统的设计 [J]. 江苏农业科学, 2014, 42 (9): 371-374.

(下转第 255 页)