

# 基于 UPPAAL 的 WSNs 数据收集协议的建模与分析

冯亚超<sup>1</sup> 杨红丽<sup>1</sup> 王 非<sup>1</sup> 武文佳<sup>1</sup> 秦胜潮<sup>1,2</sup>

(北京工业大学计算机学院 北京 100124)<sup>1</sup> (提赛德大学计算机学院 米德尔斯堡 TS1 3BA)<sup>2</sup>

**摘 要** 无线传感器网络(Wireless Sensor Networks, WSNs)广泛应用于各类数据收集系统,如居民区无线抄表(包括水表、电表和燃气表)系统。数据收集协议设计的正确性与合理性是影响网络正常运作的关键因素。针对数据收集协议的实时性需求,提出了基于 UPPAAL 实时模型检查器的 WSNs 数据收集协议的建模与分析方法。由于 UPPAAL 的输入模型相对于一般时间自动机模型而言较为复杂,因此首先对所选数据收集协议的通信行为建立一般时间自动机模型,之后再将其进一步转换为 UPPAAL 的输入模型。为了阐明该方法的有效性,选择了一个实际的无线抄表数据收集协议 WM2RP 作为例子进行建模,并利用 UPPAAL 分析其性质。分析结果显示,该协议能够满足一些与安全性及可靠性相关的性质。为了从多角度对协议进行分析,进一步建立了 WM2RP 协议的异常模型和能耗模型。

**关键词** 无线传感器网络,数据收集协议,时间自动机,UPPAAL 模型,建模与分析

**中图分类号** TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.9.024

## Modeling and Analyzing of WSNs Data Gathering Protocol Based on UPPAAL

FENG Ya-chao<sup>1</sup> YANG Hong-li<sup>1</sup> WANG Fei<sup>1</sup> WU Wen-jia<sup>1</sup> QIN Sheng-chao<sup>1,2</sup>

(College of Computer, Beijing University of Technology, Beijing 100124, China)<sup>1</sup>

(College of Computer, Teeside University, Middlesbrough TS1 3BA, UK)<sup>2</sup>

**Abstract** Wireless Sensor Networks is widely used in various types of data gathering systems, such as residential wireless meter reading (including water, electricity and gas meters) systems. The correctness and rationality of the designing of data gathering protocol are the key factors affecting the normal operation of the network. We proposed the method of modeling and analyzing of WSNs data gathering protocol based on UPPAAL towards the demand of real-time feature. Considering the input model of UPPAAL is more complex than the general terms of automata model, we established the general terms of time automata model of data gathering protocol at first, and then transferred it to the input model of UPPAAL. The effectiveness of the method is illustrated by modeling and analyzing for an actual wireless meter reading data gathering protocol WM2RP. The result shows some properties which are related to the safety and reliability can be satisfied on the protocol. The exception model and energy consumption model of WM2RP are further established to analyze the protocol from the multiple angle.

**Keywords** Wireless sensor networks, Data gathering protocol, Timed automata, UPPAAL model, Modeling and analyzing

## 1 引言

无线传感器网络方便人们轻松地获取大量与物质世界相关的各种信息,例如环境的湿度、大海的盐度等。WSNs 已被广泛应用于国防军事、国家安全、环境监测、交通管理、医疗健康以及远程抄表领域<sup>[1]</sup>。WSNs 由大量体积小、功耗低的传感器节点组成<sup>[2]</sup>。传感器节点的资源有限,具有传感、计算和通信能力,通过多跳协作的方式将采集的数据发送到汇聚节点。由于传感器节点一般由电池供电且内存有限,因此设计高效节能的 WSNs 路由协议具有重要的意义<sup>[3]</sup>。

为了使通信消耗的能量更少,路由协议经常与融合技

术<sup>[4]</sup>结合使用,融合技术可以通过减少通信量来减少通信能耗,从而延长网络的生命周期。WSNs 与传统网络的路由协议相比,具有能量优先、无全局编址、以数据为中心以及应用相关等特点<sup>[5]</sup>。

数据收集协议<sup>[6]</sup>是路由协议的一种,通信方式为多对一或一对多,用于以数据收集为主要目标的 WSNs 应用中。数据收集协议的能耗以及实时性直接影响数据收集协议的应用效果,这对系统设计至关重要,因此对数据收集协议的能耗以及实时性分析是一个重要的研究课题。

目前对数据收集协议的分析方法主要有协议仿真和协议验证两种。协议仿真是指在计算机上建立一个虚拟的网络平

到稿日期:2015-03-09 返修日期:2015-07-19

冯亚超(1987—),男,硕士生,主要研究方向为无线传感器网络、协议验证, E-mail: fengyachao1987@126.com; 杨红丽(1970—),女,博士,副教授,主要研究方向为服务计算、无线传感器网络、形式化建模;王 非(1988—),男,硕士生,主要研究方向为无线传感器网络、软件测试;武文佳(1989—),女,硕士生,主要研究方向为无线传感器网络协议;秦胜潮(1974—),男,博士,教授,主要研究方向为形式化方法、程序语言、软件工程。

台来实现对真实网络环境的模拟。技术人员在这个平台上不仅能对网络设备及网络协议进行设计研究,还能对网络的性能进行分析和评价<sup>[7]</sup>。协议验证通过使用严格的数学方法来分析协议的形式化模型,以提高系统的可靠性<sup>[8]</sup>。模型检测(Model Checking)是一种很重要的形式化方法,主要分为基于概率自动机模型和基于时间自动机模型这两种模型检查技术,其都具有广泛的应用。

概率模型检查可以对系统的不确定性进行建模,并对系统的安全性、生存性<sup>[9]</sup>进行分析与验证。时间自动机提供了一种简单高效的方法来描述具有时间因素的系统。时间自动机模型检查工具 UPPAAL<sup>[10]</sup>采用图形化的界面,适用于 Windows、Linux、SunOS 等多种操作系统,并且在一定程度上避免了状态爆炸问题。文献[11-13]主要是对模型的实时性进行形式化验证。其中,文献[13]提出了一种通用建模框架,并且可以用来对 WSNs 中的能量收集器进行建模。

## 2 WM2RP 协议介绍

WM2RP 协议是在小区抄表系统中使用的 WSNs 数据收集协议,它保证了系统在通信过程中的可靠性和实时性。下面从使用场景及体系结构这两个方面对 WM2RP 协议进行介绍。

### 2.1 协议的使用场景

WSNs 可以作为抄表系统的底层通信网络,这样的抄表系统可以完成的功能有:实时或周期性地收集表数据,收集类型可以分为单表数据收集、部分表数据收集和所有表数据收集;远程控制,管理公司可以远程控制用户家中的表,如远程关闭欠费用户家中的表、远程诊断和排除表的故障等。因此 WSNs 在抄表系统中的应用提高了抄表效率、准确性和实时性,节约了人力资源成本,减小了劳动强度。

### 2.2 协议的体系结构

WM2RP 协议是在 PEGASIS 协议<sup>[14]</sup>的基础上进行了一定程度的改进,把小区节点组织成链状,每条链最长为 35 个节点,一个小区中可能存在多条链,每条链单独管理,这样可以克服传感器节点资源不足的缺点。每个节点代表一个燃气表,燃气表中具有监测用户使用燃气情况的传感器。使用 WM2RP 协议的抄表系统的体系结构如图 1 所示。

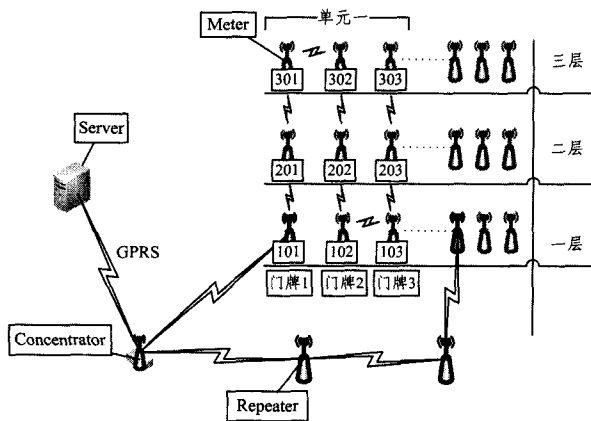


图 1 WM2RP 协议体系结构

图 1 中有 4 类节点 Server、Concentrator、Repeater 和 Meter。Server 服务器部署在厂家,由厂家维护,它与 Concentrator 的通信不属于 WSNs 的范畴,不在本文研究范围之内。

Concentrator 用于收集小区内各个用户的燃气数据,并将数据通过 GPRS 发送给 Server。每个小区只有一个 Concentrator,它的能量供应是持续的,通信范围会覆盖整个小区。Meter 是用户家中安放的燃气表,能量及通信范围都有限,并且障碍物对它的影响也很大,因此它通常被部署在靠近窗户的地方,优先与上下楼的 Meter 进行通信。由于 Meter 的通信范围有限,当 Concentrator 的位置与 Meter 的距离超过 Meter 的通信范围时,就需要部署 Repeater 来进行中转。Repeater 与 Meter 使用相同的设备,只是在网络中承担的职责不同,它不检测用户的燃气使用情况,只承担路由的责任。

一次完整的抄表过程为:Server 首先通过 GPRS 发送命令给 Concentrator,Concentrator 收到命令后再将命令转发给 Meter,Meter 根据协议链将抄表命令逐级传送,在协议链中最后一个作为叶子节点的 Meter 将响应命令的数据向上逐级返回,直至返回到 Concentrator,Concentrator 再通过 GPRS 将数据返回给 Server。

## 3 数据收集协议的建模与分析方法

WM2RP 协议应用在燃气抄表系统中,简单易操作,并且具有很好的可靠性,在抄表系统中发挥了重要的作用。然而 WM2RP 协议也有很多不足,例如该协议并没有形式化模型,只有自然语言描述的一个规范,工程师只能根据规范凭借经验来对协议进行实现,这样十分不严谨。因此需要对协议建立更加严谨的形式化模型,并通过模型检查的方法证明协议设计的正确性。

本节首先介绍时间自动机的定义,然后在此基础上提出针对 WSNs 数据收集协议的建模与分析方法。

### 3.1 时间自动机

时间自动机<sup>[15]</sup>是对自动机理论的扩展,提供了形式化的方法来建立和分析实时系统的行为,在模型检测方面有着重要的应用。

定义 1 时间自动机是一个六元组  $(L, l_0, C, A, E, I)$ 。其中, $L$  代表有限的位置集合; $l_0 \in L$  代表初始位置; $C$  代表时钟(Clock)集合; $A$  代表动作(Action)集合; $E \subseteq L \times B(C) \times A \times 2^C \times L$  代表边集合, $B(C)$  表示时钟约束集合, $2^C$  表示时钟重置集; $I: L \rightarrow B(C)$  是一个映射,它把  $L$  中的每个位置映射到  $B(C)$  中的某个时钟约束,表示位置的不变式,是对位置进行约束,并且给出了迁移出位置的时间约束条件。

### 3.2 建模与分析方法

本文针对数据收集协议的行为受时间约束的特点,提出了数据收集协议的建模与分析方法,如图 2 所示。

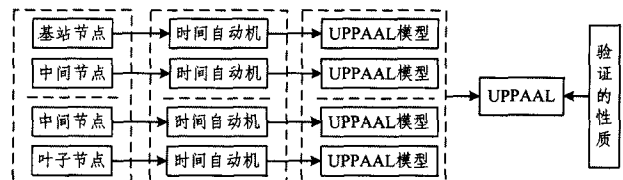


图 2 数据收集协议的建模与分析方法

由于时间自动机模型只关注节点之间的通信交互行为与时钟约束,不考虑与实现相关的变量和函数定义等具体细节,模型相对简单,且更容易理解。因此首先为数据收集协议中的各类节点(包括基站、中间节点和叶子节点)建立时间自动机模型。为了利用 UPPAAL 工具对协议进行分析,还需要

进一步将建立好的时间自动机模型转换为 UPPAAL 的输入模型。

由于 UPPAAL 的输入模型相对复杂,包括迁移时间约束、状态停留时间约束、时间自动机的交互通信等,普通用户很难直接准确建立 UPPAAL 的输入模型。本文提出的两阶段建模方法降低了建模难度,可以方便、快速、正确地对协议行为进行建模。

#### 4 WM2RP 协议的建模与分析

本节首先以 WM2RP 协议的中间节点为例,介绍了时间自动机模型的建立过程,接着介绍了时间自动机模型与 UPPAAL 模型的转换关系以及转换后的 UPPAAL 模型。最后对 WM2RP 协议的性质进行分析。

##### 4.1 WM2RP 协议中间节点的时间自动机模型

在 WM2RP 协议中,根据节点承担角色的不同将其划分为 3 类:基站、中间节点和叶子节点,每类节点的行为既有相同之处也有不同之处,因此需要为每类节点分别建立模型。下面以中间节点为例介绍 WM2RP 协议时间自动机模型的建立过程。

中间节点是由电池供电,能量有限。本文工作主要研究节点的实时特性,因此只对节点中影响时间的行为进行建模。中间节点的时间自动机模型如图 3 所示。

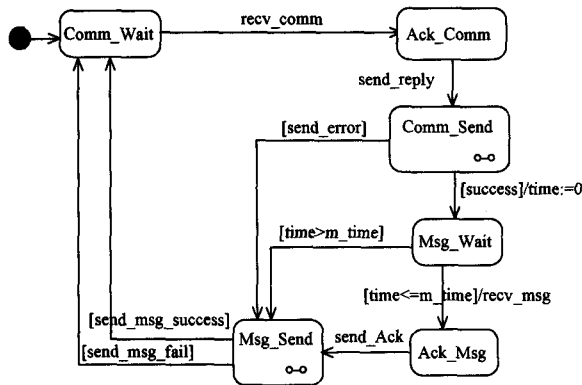


图 3 中间节点的时间自动机模型

(1)中间节点在等待接收命令状态 Comm\_Wait 收到命令后就会进入发送确认收到命令状态 Ack\_Comm。

(2)在发送确认收到命令状态 Ack\_Comm,中间节点向

父节点发送 ACK 后就会进入发送命令状态 Comm\_Send。

(3)发送命令状态 Comm\_Send 是一个复合状态,如果发送命令失败就会跳转到发送数据状态 Msg\_Send,发送成功后就会跳转到等待接收数据状态 Msg\_Wait。

(4)在发送数据状态 Msg\_Wait,中间节点会等待  $m\_time$  时间,在  $m\_time$  时间内收到下级节点发送的数据后,就会跳转到发送确认收到数据状态 Ack\_Msg,等待超时则会跳转到发送数据状态 Msg\_Send。

(5)在发送收到数据确认状态 Ack\_Msg,中间节点准备向下级节点发送 ACK,发送完 ACK 后就跳转到发送数据状态 Msg\_Send。

(6)发送数据状态 Msg\_Send 是一个复合状态,中间节点如果在发送命令失败或者等待接收数据超时进入这个状态,就会自己填充数据向父节点发送数据。

##### 4.2 时间自动机模型与 UPPAAL 模型的转换关系

为了利用 UPPAAL 工具对协议进行分析,需要将 WM2RP 协议的时间自动机模型进一步转换为 UPPAAL 的输入模型。该转换对应的关系包括:

(1)在时间自动机模型中具体的动作或者标记对应 UPPAAL 模型中的动作。

(2)在 UPPAAL 模型中,动作包括:

1)接收动作:  $send[n]?, reply[n]?, rcv[n]?, rp\_data[n]?$ 。 $n$  代表节点的 ID,  $send[n]?$  代表节点  $n$  等待接收命令,  $reply[n]?$  代表节点  $n$  等待接收确认收到命令消息,  $rcv[n]?$  代表等待接收数据,  $rp\_data[n]?$  代表等待接收确认收到数据消息。

2)发送动作:  $send[n]!, reply[n]!, rcv[n]!, rp\_data[n]!$ 。 $n$  为动作的目标对象的 ID,  $send[n]!$  代表向节点  $n$  发送命令,  $reply[n]!$  代表向节点  $n$  发送确认收到命令消息,  $rcv[n]!$  代表向节点  $n$  发送数据,  $rp\_data[n]!$  代表向节点  $n$  发送确认收到数据消息。

3)更新动作:  $S_i U(n)$ 。在 UPPAAL 模型中,  $S_i$  和  $S_j$  代表位置,  $S_i U(n)$  就是从位置  $S_i$  到  $S_j$  跳转时执行的更新动作(以  $U(Update)$  结尾以便区分)。

4)约束条件:  $S_i G(n), time \leq t$ 。  $S_i G(n)$  是从位置  $S_i$  到  $S_j$  跳转必须满足的条件(以  $G(Guard)$  结尾以便区分)。

时间自动机模型到 UPPAAL 模型的转换关系如表 1 所列。

表 1 时间自动机模型到 UPPAAL 模型的转换关系

| 序号 | 时间自动机模型                                | UPPAAL 模型                                | 说明   |
|----|--|--|--|
| 1  | $(S_0 \xrightarrow{rcv\_comm} S_1)$    | $S_0 \xrightarrow{send[n]?} S_1$         | 在做转换时,UPPAAL 模型对应到一个具体的节点,其标识为 $n$ 。 $send[n]?$ 代表接收命令。  |
| 2  | $(S_0 \xrightarrow{scomm(x)} S_1)$     | $S_0 \xrightarrow{send[nNode[n]]!} S_1$  | 在 UPPAAL 模型中, $nNode[n]$ 为节点 $n$ 的子节点,这是节点 $n$ 向子节点 $nNode[n]$ 发送命令。                                 |
| 3  | $(S_0 \xrightarrow{send\_ack} S_1)$    | $S_0 \xrightarrow{reply[pNode[n]]!} S_1$ | $pNode[n]$ 为节点 $n$ 的父节点。这是节点 $n$ 向父节点 $pNode[n]$ 发送确认收到命令的转换。  |
| 4  | $(S_0 \xrightarrow{rcv\_ack} S_1)$     | $S_0 \xrightarrow{reply[n]?} S_1$        | 节点 $n$ 接收确认收到命令的 ACK。  |
| 5  | $(S_0 \xrightarrow{tm:=tm+1} S_1)$     | $S_0 \xrightarrow{S_0!U} S_1$            | $tm$ 统计超时次数, $S_0!U(n)$ 是 UPPAAL 模型下的函数, $n$ 是节点的 ID。这个转换就是更新超时次数,会将其放到函数名以 $U$ 为后缀的 $S_0!U(n)$ 函数中。 |
| 6  | $(S_0 \xrightarrow{time \leq 3s} S_1)$ | $S_0 \xrightarrow{time[n] < WRT+1} S_1$  | $WRT$ 是 UPPAAL 中的常量,在时间上为 $3s$ 。   |
| 7  | $(S_0 \xrightarrow{tm \geq 3} S_1)$    | $S_0 \xrightarrow{S_0!G(n)} S_1$         | $tm \geq 3$ 约束会放到 UPPAAL 模型的函数名以 $G$ 为后缀的函数里。 $n$ 代表当前节点的 ID。  |

### 4.3 WM2RP 协议的 UPPAAL 模型

WM2RP 协议的 UPPAAL 模型包括基站模型、中间节点模型、叶子节点模型以及全局声明(全局变量和全局函数)。下面以中间节点为例介绍 WM2RP 协议的 UPPAAL 模型。

中间节点的 UPPAAL 模型通过管道变量(chan)与相邻节点(包括基站和叶子节点)进行同步。中间节点的位置集包含基站和叶子节点的位置集。中间节点的 UPPAAL 模型如图 4 所示。

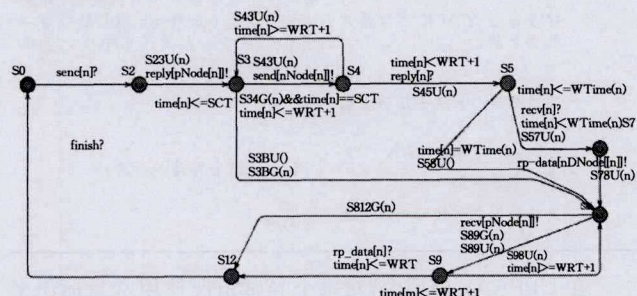


图 4 中间节点的 UPPAAL 模型

(1) 等待接收命令位置 S0: 这是节点的初始位置, 在 UPPAAL 中用双圆表示。在此位置可以停留任意时间,  $send[n]?$  一旦被触发就会跳转到位置 S2。

(2) 发送收到命令确认位置 S2: 位置 S2 是一个紧急位置, 在此位置不能有时间上的延迟, 中间节点立即向其父节点发送收到命令信号的 ACK, 然后跳转到位置 S3。

(3) 发送命令位置 S3: 中间节点在 S3 位置延迟 SCT 时间, 然后检查当前节点的子节点  $nNode[n]$  是否有效。子节点有效就会向其发送命令, 并跳转到位置 S4; 如果子节点无效, 就跳转到位置 S8。

(4) 等待命令确认位置 S4: 中间节点在该位置等待接收子节点发送的 ACK, 等待时间为 WRT。在 WRT 时间内, 中间节点如果收到 ACK, 就会跳转到位置 S5; 如果没有收到 ACK, 则跳转到位置 S3。在从位置 S4 跳转到位置 S3 时会根据超时次数来更新子节点  $nNode[n]$ 。

(5) 等待接收数据位置 S5: 中间节点在该位置等待 WTime(n) 时间。在 WTime(n) 时间内, 中间节点收到消息就跳转到位置 S7, 没有收到数据就跳转到位置 S8。

(6) 发送 ACK 位置 S7: 节点  $nDNNode[n]$  是给节点 n 发送数据的源节点。位置 S7 是一个紧急位置, 在此位置不能有时间上的延迟, 中间节点会立即发送 ACK, 并跳转到位置 S8。

(7) 发送数据位置 S8: 中间节点首先检查父节点是否在通信的范围内, 如果在就向其父节点发送数据, 并跳转到位置 S9; 如果父节点不在其通信范围内, 则跳转到位置 S12。

(8) 等待接收数据确认位置 S9: 中间节点在该位置等待 WRT 时间, 如果在 WRT 时间内, 中间节点收到 ACK 就进入位置 S12; 没有收到 ACK, 则跳转到位置 S8 并增加超时次数, 如果超时次数达到 3, 表明当前节点异常, 就会更新自己的父节点, 更新方式为设置自己的父节点为父节点的父节点。

(9) 结束位置 S12: 节点 n 跳转到位置 S12 时, 表明在整个数据收集过程中节点 n 完成自己担负的任务。

### 4.4 WM2RP 协议的性质与分析

本节首先介绍 WM2RP 协议中的一些与安全性和可靠性相关的性质, 然后利用 UPPAAL 工具对这些性质进行验证。

对于 WM2RP 协议而言, 主要是验证时间等待模型是否可以正常使用, 并使用全局变量 fail\_count 来表示是否有数据收集失败和发送命令失败的节点。

**性质 1(死锁)** 时间自动机模型中的死锁是指模型在运转过程中可能会到达一个没有后继状态的状态, 从而永远停留在该状态。对于 WM2RP 协议而言, 如果模型中存在死锁, 则说明协议的设计存在问题, 可能会导致以该协议为基础的抄表系统在运作过程中因某一特殊情况的出现而永远“卡死”。

在 UPPAAL 工具中, 检验模型中存在死锁的性质描述为“E <> deadlock”, 对应的检验模型中不存在死锁的性质描述为“A [] not deadlock”。

利用 UPPAAL 工具验证 WM2RP 协议是否不存在死锁的结果如图 5 所示, 结果为满足性质, 即该模型不存在死锁。

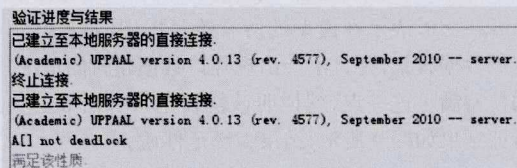


图 5 死锁的验证结果

**性质 2(安全性)** 安全性是指在时间自动机模型的一条路径上, 所有状态都满足要验证的性质。安全性有两种情况: 1) 一条路径上的所有状态都满足某个性质, 在 UPPAAL 工具中用“E [] P”来描述, 其图形表示如图 6(a) 所示, 其中 P 代表某个特定的性质; 2) 所有路径上的每一个状态都满足某个条件, 在 UPPAAL 工具中对应的查询语言为“A [] P”, 其图形表示如图 6(b) 所示。

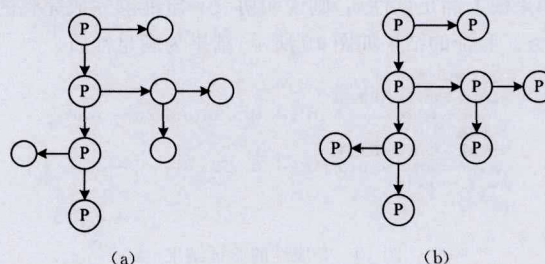


图 6 安全性

对于 WM2RP 协议而言, 属性公式“E [] fail\_count == 0”表示存在一条路径, 在该路径中变量 fail\_count 的值始终为 0, 即不会有节点数据收集失败。如果协议的模型能够满足该性质, 则说明协议至少存在一种情况可以正常完成抄表, 因此是安全的。在 UPPAAL 工具中验证该性质的结果如图 7 所示, 结果为满足性质, 即存在一条路径, 使得在该路径上的每个状态中都满足“fail\_count == 0”。

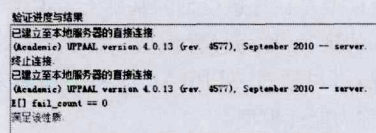


图 7 安全性的验证结果

**性质 3(活性)** 活性是指在时间自动机模型的所有路径上, 一定存在一个满足特定条件的状态。活性有两种情况: 1) 在所有路径上, 将来一定会有一个状态满足条件 P, 对应的 UPPAAL 工具中的查询语言为“A <> P”, 图形表示如图 8(a)

所示;2)在某个状态满足条件  $q$  时,所有以此状态为起始状态的路径上,将来一定有某个状态满足条件  $P$ ,对应的 UPPAAL 工具中的查询语言为“ $q \rightarrow P$ ”,其图形表示如图 8(b)所示。

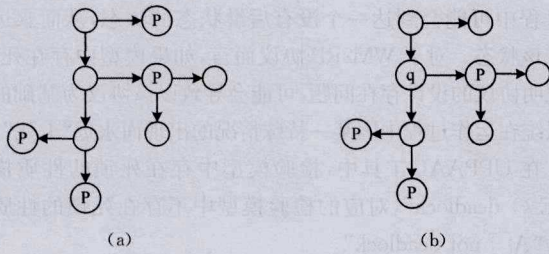


图 8 活性

对于 WM2RP 协议而言,属性公式“ $A \langle \rangle \text{base.Succ} \ \&\& \ \text{fail\_count} = 0$ ”表示在模型的所有路径中,基站最终都会成功地完成数据收集,并且不会出现等待数据超时的情况,如果协议的模型满足这一点,就说明该协议是具有活性的。该性质的验证结果如图 9 所示,结果为满足性质。

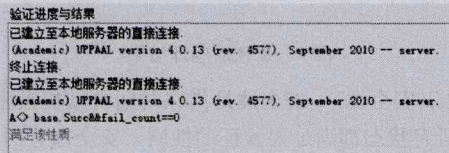


图 9 活性的验证结果

性质 4 (完成性) 对于 WM2RP 协议而言,属性公式“ $\text{base.S0} \rightarrow \text{base.Succ}$ ”表示当基站启动后,一定能够完成数据收集,最终到达位置 Succ 且没有等待接收数据失败的情况发生,如果模型满足该性质,则说明协议一定能够完成完整的抄表任务。验证的结果如图 10 所示,结果为满足性质。

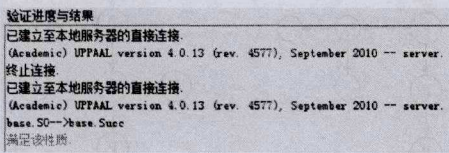


图 10 完成性的验证结果

## 5 WM2RP 协议的扩展模型

工业界将协议实际应用到具体的抄表系统时,通常关心的是协议的可靠性、实时性以及能耗情况,因此本文针对这一问题建立了 3 个不同的模型,使得在对协议进行分析时可以根据不同的需求使用不同的模型。第 1 个模型建立在协议的理想情况下,称为理想模型,在第 4 节中已经进行了详细的介绍。以理想模型作为基础,可以根据特定需求来对其进行修改。第 2 个模型是在理想模型的基础上考虑了节点的各种异常情况带来的延时,称为异常模型。第 3 个模型考虑了节点的能耗情况,可以通过 UPPAAL 的模拟功能来对节点能耗进行评估,称为能耗模型。

### 5.1 WM2RP 协议的异常模型

WM2RP 协议的异常模型是在理想模型的基础上进一步考虑了各节点的各种异常情况。节点发生异常会延长整个数据收集过程消耗的时间。在基站的初始化函数里可以对节点的异常情况进行设置,节点所有可能发生的异常情况如表 2 所列。

表 2 节点的异常情况

| 序号 | 异常情况                   | 延时触发的情况  |
|----|------------------------|--|
| 1  | 节点接收命令之前               | 会触发上级节点连续 3 次发送命令,并且等待 ACK                               |
| 2  | 节点接收命令之后发送响应命令的 ACK 之前 | 会触发上级节点连续 3 次发送命令,并且等待 ACK                               |
| 3  | 发送响应命令的 ACK 之后发送命令之前   | 触发上级节点一直等待数据,并会超时  |
| 4  | 发送命令之后,接收命令的 ACK 之前    | 触发上级节点一直等待数据,并会超时  |
| 5  | 接收命令的 ACK 之后接收数据之前     | 会触发下级节点发送数据失败,如果当前节点是下级节点能通信的最后一个节点,那么会触发当前节点的上级节点等待数据超时 |
| 6  | 接收数据后发送响应数据的 ACK 之前    | 同上   |
| 7  | 发送响应数据的 ACK 之后发送数据之前   | 导致上级节点等待数据超时   |
| 8  | 发送数据之后接收响应数据的 ACK 之前   | 无影响  |

在 UPPAAL 模型中,对每个异常情况使用常量标记来表示,图 11 描述的是 UPPAAL 模型中对应节点异常情况的常量标记代码。aberN 数组代表每个节点(不包括基站)的当前工作状态。

```

//1-7代表异常, 0代表正常
int aberN[NC-1];
const int RC = 1; //接收命令之前
const int SCA = 2; //发送接收命令的ack之前, 接收命令之后
const int SC = 3; //发送命令之前, 发送接收命令的ack之后
const int RCA = 4; //接收命令ack之前, 发送命令之后
const int RD = 5; //接收数据前, 接收命令ack之后
const int SDA = 6; //发送响应数据的ack之前, 接收数据之后
const int SD = 7; //发送响应数据的ack之后, 发送数据之前

```

图 11 异常情况常量标记

节点的异常模型需要在每个状态转换中检查在当前状态是否异常,如果异常就会跳转到专门用来表示节点异常的状态 S13。中间节点的异常模型如图 12 所示。

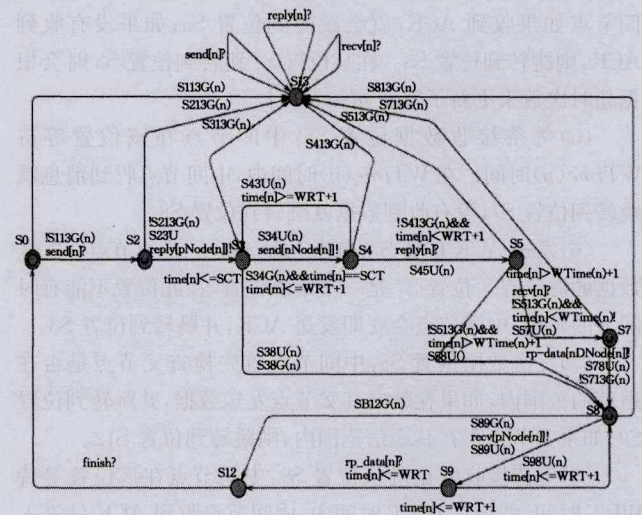


图 12 中间节点的异常 UPPAAL 模型

异常模型主要用来验证在异常的情况下基站是否可以收集到其他正常节点的数据。节点出现异常会增加整个协议链中数据收集的时间,但是基站等待接收数据的时间是固定的,如果节点异常导致的延时过长,基站就会在后面的节点还在继续收集数据时发生等待数据超时,从而无法收集到正常节点的数据。针对异常模型,同样利用 4.4 节中的性质对其进行验证,结果如图 13 所示。

```

验证进度与结果
已建立至本地服务器的直接连接.
(Academic) UPPAAL version 4.0.13 (rev. 4577), September 2010 -- server.
终止连接
已建立至本地服务器的直接连接.
(Academic) UPPAAL version 4.0.13 (rev. 4577), September 2010 -- server.
A[] not deadlock
满足该性质
E[] fail_count==0
满足该性质
A[] fail_count == 0
满足该性质
A◇ base_Succ
满足该性质
base.S0-->base_Succ
满足该性质

```

图 13 异常模型验证结果

## 5.2 WM2RP 协议的能耗模型

协议的能耗情况是影响网络生命周期的首要因素。把基站对协议链中所有节点的数据进行一次收集的过程称为一轮,则生命周期可以用数据收集轮数来衡量。在 UPPAAL 中,可以对协议中节点之间的交互行为进行模拟,并根据交互类型的不同来设置消耗一定的能量。

在理想模型的基础上添加的与能量相关的全局变量如图 14 所示。

```

const int ENERGY_INIT = 10000; //初始能量值
int energyInit[NC]; //每个节点的能源存储

const int MIN_SND_RE_ENERGY = 10; //发送一个ack最小的能量
const int MIN_RCV_RE_ENERGY = 5; //接收一个ack需要最小的能量
const int MIN_SND_DATA_ENERGY = 30; //发送数据消息的需要的最小能量
const int MIN_RCV_DATA_ENERGY = 15; //接收数据消息需要的最小能量
const int MIN_RECV_COMM_ENERGY = 10; //接收命令需要的最小能量
const int MIN_SND_COMM_ENERGY = 20; //发送命令需要的最小能量

const int ECSR = 1;
//能耗类型 发送ack的能量消耗类型-ENERGY_COST_SEND_REPLY_TYPE
const int ECSM = 5;
//能耗类型 发送数据的时候消耗的能量的类型
const int ECRM = 6;
//能耗类型 接收数据时消耗的能量的类型
const int ECCR = 2;
//能耗类型 接收响应命令的ack的能量消耗消息-ENERGY_COST_RECV_REPLY_TYPE
const int ECRD = 3;
//能耗类型 接收命令能量消耗类型-ENERGY_COST_RECV_DATA_TYPE
const int ECSD = 4;
//能耗类型 发送命令能量消耗类型-ENERGY_COST_SEND_DATA_TYPE

```

图 14 能耗相关的全局变量

在理想模型的基础上添加的与能量相关的全局函数如图 15 所示。

```

void EnergyCost(int node, int type)
{
    if (ECSR == type) //节点发送ack后, 消耗能量
    {
        if (energyInit[node] >= MIN_SND_RE_ENERGY)
            energyInit[node] = energyInit[node] - MIN_SND_RE_ENERGY;
    }
    if (ECCR == type) //节点接收ack后, 消耗能量
    {
        if (energyInit[node] >= MIN_RCV_RE_ENERGY)
            energyInit[node] = energyInit[node] - MIN_RCV_RE_ENERGY;
    }
}

bool EpyEgh(int node, int type)
{
    if (ECSR == type)
    {
        //检查节点node的能量是否足够发送ack
        if (energyInit[node] < MIN_SND_RE_ENERGY)
            return false;
        else
            return true;
    }
}

```

图 15 能耗相关的全局函数

能耗模型中的全局常量主要用来表示节点的动作类型以及相应的动作需要消耗能量的多少,全局函数则根据节点的交互类型来更新表示能量的变量。中间节点的能耗模型如图 16 所示。

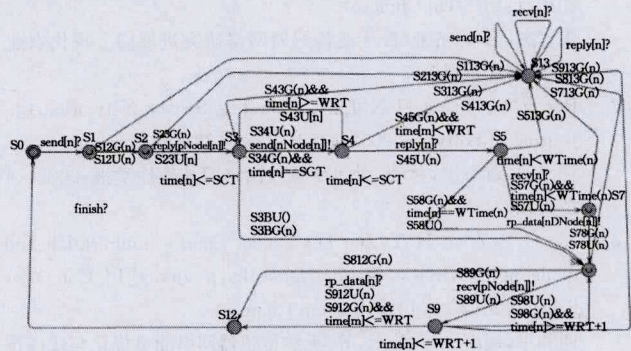


图 16 中间节点的能耗模型

在位置 S1—S9 这 9 个状态都会检查能量是否足以支持交互行为,如果不支持就会跳转到位置 S13。

利用 UPPAAL 工具的模拟功能,模拟节点间的交互行为来完成一轮数据收集,从而可以观察到一轮数据收集后各个节点的能量消耗情况。设能量消耗最多的节点消耗的能量值为 MAX\_SING\_COST,然后使用初始能量 ENERGY\_INIT 就可以计算出数据收集轮数  $N = ENERGY\_INIT / MAX\_SING\_COST$ 。UPPAAL 数据收集模拟的过程如图 17 所示,这是 8 个节点的协议链。

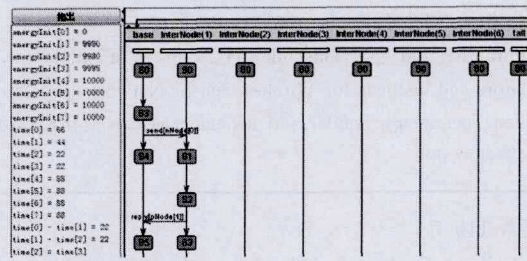


图 17 UPPAAL 数据收集模拟过程

**结束语** 本文提出了基于 UPPAAL 实时模型检查器的 WSNs 数据收集协议的建模与分析方法。首先为所协议建立相对简单的时间自动机模型,再将其进一步转化为可以作为 UPPAAL 工具输入的 UPPAAL 模型。为了阐明该方法的有效性,选择了一个实际的无线抄表数据收集协议 WM2RP 作为例子进行了建模,并利用 UPPAAL 分析了一些与安全性和可靠性相关的性质。为了从多角度对协议进行分析,进一步建立了 WM2RP 协议的异常模型和能耗模型,并进行了相应的验证与模拟。

进一步工作:对 WM2RP 协议的 UPPAAL 模型进行优化。目前该模型的状态空间很大,一种方案是利用对时间自动机网络做积的方式来缩小状态空间;另一种方案是对目前该协议的 UPPAAL 模型中的状态进行精化,从而达到缩减状态空间的目的。另一方面,还可以使用概率时间自动机对 WM2RP 协议进行建模。因为在仅使用时间自动机模型验证协议的可靠性时,只是验证各种异常情况是否可以收集到其他正常节点的数据,然而这些异常情况在实际使用时发生的可能性是一个随机因素,因此可以从概率的角度入手对协议进行较为全面的评估。

## 参考文献

- [1] Zhang F J, Peng Z, Qi Y F. Research progress on Wireless Sensor Networks[J]. Modern Agricultural Science and Technology, 2010, 1: 16-17 (in Chinese)  
张付杰, 彭争, 齐亚峰. 无线传感器网络研究进展[J]. 现代农业科技, 2010, 1: 16-17
- [2] Ren F Y, Huang H N, Lin C. Wireless Sensor Networks[J]. Journal of Software, 2003, 14(7): 1282-1291 (in Chinese)  
任丰原, 黄海宁, 林闯. 无线传感器网络[J]. 软件学报, 2003, 14(7): 1282-1291
- [3] Xie H P, Zhou H Y, Zuo D C, et al. Energy optimization and modeling in Wireless Sensor Networks: a survey[J]. Computer Science, 2012, 39(10): 15-20 (in Chinese)  
谢和平, 周海鹰, 左德承, 等. 无线传感器网络能量优化与建模技术综述[J]. 计算机科学, 2012, 39(10): 15-20
- [4] Li L, Li J. Research of compressed sensing theory in WSN data fusion[C]// Fourth International Symposium on Computational Intelligence and Design. Hangzhou, China: IEEE, 2011, 2: 125-128
- [5] Akkaya K, Younis M. A survey on routing protocols for Wireless Sensor Networks[J]. Ad Hoc Networks, 2005, 3(3): 325-349
- [6] Ma C M, Wang W L, Hong Z. Improved Energy Efficient Data Gathering Protocol in Wireless Sensor Network[J]. Computer Science, 2015, 42(2): 65-69 (in Chinese)  
马晨明, 王万良, 洪榛. 无线传感器网络中一种改进的能效数据收集协议[J]. 计算机科学, 2015, 42(2): 65-69
- [7] Imran M, Said A M, Hasbullah H. A survey of simulators, emulators and testbeds for Wireless Sensor Networks[J]. Information Technology, Bandar Seri Iskandar, Malaysia: IEEE, 2010, 6(2): 897-902
- [8] Elleuch M, Hasan O, Tahar S, et al. Formal analysis of a scheduling algorithm for Wireless Sensor Networks[C]// 13th International Conference on Formal Engineering Methods, ICFEM 2011. Durham, UK, 2011: 388-403
- [9] Zhou Q L, Zhang B, Xi L. System survivability analysis based on model checking[J]. Computer Engineering, 2012, 38(17): 38-41 (in Chinese)  
周清雷, 张兵, 席琳. 基于模型检测的系统生存性分析[J]. 计算机工程, 2012, 38(17): 38-41
- [10] Behrmann G, David A, Larsen K G. A tutorial on Uppaal[M]// Formal Methods for the Design of Real-Time Systems; Lecture Notes in Computer Science. 2004: 200-236
- [11] Guo Z L, Gao C H, Ma L C, et al. Formal Verification of Safety Computer Platform Based on Timed Automata Model[J]. Journal of the China Railway Society, 2011, 33(6): 69-73 (in Chinese)  
郭志良, 郗春海, 马连川, 等. 基于时间自动机模型的安全计算机平台的形式化验证[J]. 铁道学报, 2011, 33(6): 69-73
- [12] Saghar K, Henderson W, Kendall D, et al. Applying formal modelling to detect DoS attacks in wireless medium[C]// CSNDSP. 2010
- [13] Wu N. Modelling and analysis of Wireless Sensor Networks with energy harvesting capabilities[C]// Informatics and Mathematical Modelling. 2012
- [14] Heinzelman W, Chandrakasan A, Balakrishnan H. Energy-Efficient communication protocol for wireless micro sensor networks[C]// Proc. of the 33rd Annual Hawaii Int'l Conf. on System Sciences. Maui: IEEE, 2000: 3005-3014
- [15] Bengtsson J, Wang Y. Timed automata: semantics, algorithms and tools[M]// Lectures on Concurrency and Petri Nets; Advances in Petri Nets. 2004: 87-124
- 
- (上接第 110 页)
- [9] Gavilanes G, Quercia R D, Jaimes A. Cultural dimensions in twitter: Time, individualism and power[C]// Proceedings of the 7th International Conference on Weblogs and Social Media. Boston: AAAI, 2013: 195-204
- [10] Bakhshi S, Kanuparth P, Gilbert E. Demographics, weather and online reviews: A study of restaurant recommendations[C]// Proceedings of the 23rd International World Wide Web Conference. New York: ACM, 2014: 443-454
- [11] Hu Ya-hui, Yu Wei, Li Shi-jun, et al. CFM: A fast rating predicting method based on the culture and factorization machine [J]. Journal of Nanjing University, 2015, 51(4): 826-833 (in Chinese)  
胡亚慧, 李石君, 余伟, 等. CFM: 一种结合文化和因子分解机的快速评分预测方法[J]. 南京大学学报, 2015, 51(4): 826-833
- [12] Meng Xiao-feng, Li Yong, Zhu Jian-hua. Social computing in the era of big data: Opportunities and challenges [J]. Journal of Computer Research and Development, 2013, 50(12): 2483-2491 (in Chinese)  
孟小峰, 李勇, 祝建华. 社会计算: 大数据时代的机遇与挑战[J]. 计算机研究与发展, 2013, 50(12): 2483-2491
- [13] Yu Wei, Li Shi-jun, Yang Sha, et al. Automatically discovering of inconsistency among cross-source data based on Web big data [J]. Journal of Computer Research and Development, 2015, 52(2): 295-308 (in Chinese)  
余伟, 李石君, 杨莎, 等. Web 大数据环境下的不一致跨源数据发现[J]. 计算机研究与发展, 2015, 52(2): 295-308
- [14] Hu Ya-hui, Li Shi-jun, Yu wei, et al. Recognizing the same commodity entities in big data [J]. Journal of Computer Research and Development, 2015, 52(8): 1794-1805 (in Chinese)  
胡亚慧, 李石君, 余伟, 等. 大数据环境下的电子商务商品实体同一性识别[J]. 计算机研究与发展, 2015, 52(8): 1794-1805
- [15] Tsai Chihfong, Hung Chihli. Cluster ensembles in collaborative filtering recommendation [J]. Applied Soft Computing, 2012, 12(4): 1417-1425
- [16] Srebro N, Rennie J, Jaakola T. Maximum-margin matrix factorization [J]. Advances in Neural Information Processing Systems, 2005, 17: 1329-1336
- [17] Kolda T G, Bader B W. Tensor decompositions and applications [J]. Society for Industrial and Applied Mathematics, 2009, 51(3): 455-500
- [18] Rendle S. Factorization Machines with libFM [J]. ACM Transactions on Intelligent Systems and Technology, 2012, 3(3): 219-224