Software Engineering Group
Department of Computer Science
Nanjing University
http://seg.nju.edu.cn

Technical Report No. NJU-SEG-2013-CJ-001

2013-CJ-001

基于随机时间自动机和统计模型检验技术的无线 传感网络协议建模与分析

张凤玲，卜磊，王林章，赵建华，李宣东

中国科学: 信息科学 2013 年 第 43 卷 第 1 期: 90–107

中国科学: 信息科学 2013 年 第 43 卷 第 1 期: 90–107
www.scichina.com info.scichina.com
《中国科学》杂志社
SCIENCE CHINA PRESS
互联网计算的软件技术专刊 · 论文

基于随机时间自动机和统计模型检验技术的无线传感网络协议建模与分析

张凤玲, 卜磊*, 王林章, 赵建华, 李宣东

南京大学计算机软件新技术国家重点实验室, 南京 210023
* 通信作者. E-mail: bulei@nju.edu.cn

收稿日期: 2012–08–06; 接受日期: 2012–11–09
国家自然科学基金 (批准号: 61100036, 61021062, 61170066) 和国家高技术研究发展计划 (批准号: 2011AA010103, 2012AA011205) 资助项目

摘要 近年来, 传感器技术得到了长足而有效的提升, 无线传感网络 (WSN) 以其开放、动态的特征获得了极大的关注, 并成为了互联网计算的一个重要组成. WSN 系统行为复杂, 经常面临信息丢失、节点动态变化等不确定因素, 且网络中的节点一旦部署将很难更改、维护. 因此, 为了保证相关应用的正常工作, 在系统设计阶段对 WSN 中的底层协议进行质量保障就成为了一项非常重要的研究问题. 系统设计人员不仅需要保证协议功能上的正确性, 还应该评估协议在目标工作环境下的性能, 以保证其可以胜任相应的工作需求.

针对以上问题, 本文提出了一种基于随机时间自动机和统计模型检验技术的 WSN 协议建模、分析和评估途径. 在建模阶段, 首先将采用时间自动机对协议在理想环境下的基本业务流程进行建模. 考虑到 WSN 系统实际工作中会遇到的各种不确定性因素, 将用带权分枝来对模型进行扩展, 生成协议的随机时间自动机. 在验证阶段, 首先采用经典模型检验技术, 在理想时间自动机上检验相关功能性质, 保证协议工作逻辑的正确性. 为评估协议在不同条件下的具体性能, 则在随机时间自动机上用统计模型检验技术对其进行数值分析, 以进行参数配置、性能预测、协议比较等工作. 为展示该途径的可用性及其技术细节, 本文对两种著名的 WSN 时间同步协议, TPSN 和 FTSP 分别进行了完整的建模与评估.

关键词 无线传感网络协议 时间自动机 随机时间自动机 模型检验 基于统计的模型检验

1 引言

随着计算机网络技术的快速发展, 互联网计算成为了 21 世纪信息技术发展的一个重要方向. 近年来, 传感器技术得到了长足而有效的提升. 相应的, 无线传感网络技术 (wireless sensor network, WSN)[1] 以其开放、动态的特征得到了极大的关注, 并成为了互联网计算的一个重要组成. 随着 WSN 技术的普及, 大规模的 WSN 网络越来越广泛地被应用于医疗、军事、环境监测和保护等安全攸关领域, 因此对其服务质量进行可信保障是一项刻不容缓的工作. 在 WSN 系统中, 大量传感节点协同工作进行数据的收集、处理与信息交换, 以感应和监测其周围环境. 为使 WSN 中的高层应用能够正确、高效、

引用格式: 张凤玲, 卜磊, 王林章, 等. 基于随机时间自动机和统计模型检验技术的无线传感网络协议建模与分析. 中国科学: 信息科学, 2013, 43: 90–107, doi: 10.1360/112012-498

可信地进行, 必须保证其底层协议的正确性. 此外, WSN 一般工作在复杂环境中, 工作环境中经常会面临环境干扰、动态网络等不确定性影响, 并且一旦部署完成, 难以对网络进行维护和更改. 因此, 在 WSN 系统部署之前对相关底层协议进行完整的验证与评估至关重要.

目前, 检验协议正确性的方法主要包括仿真、测试和形式化验证. 仿真和测试可用于检验大规模系统, 并发现其中存在的明显错误. 但这两种方法无法测试系统的所有行为, 不能发现系统在一些特殊场景下可能出现的错误, 因此无法保证已测试系统的正确性. 为了在设计阶段尽可能多的发现协议中存在的错误, 保证协议的正确性, 大量研究者使用形式化方法来对系统行为进行建模与验证 [2].

由于 WSN 协议与时间密切相关, 时间自动机 (timed automata)[3,4] 是对其建模的理想选择. 时间自动机提供了完善的时间建模机制, 可以进行时钟之间的比较、重置等, 以建模协议中对信息传输、信息延迟等的时间限制, 其同步信息则可模拟 WSN 中信息的发送和接收. 时间自动机的模版机制则为大规模系统建模提供了良好的可扩展性. 因此, 使用时间自动机方便地建立协议在理想环境下的工作流程模型. 此外, 为了分析 WSN 协议在真实环境中的工作状态, 在建模阶段必须考虑环境中存在的信息丢失、节点失效等不确定性因素, 而现有工作对此方面鲜有触及, 经典时间自动机也无法描述并表达不确定行为. 针对该问题, 本文提出利用随机时间自动机 (stochastic timed automata)[5], 在模型中引入带权分枝来表达系统在各状态间的随机转移, 建模并描述系统中的不确定性因素.

模型检验 (mode checking)[6] 是形式化验证的一种常用方法. 传统的模型检验通过遍历系统的所有状态空间来验证系统是否满足某一性质, 可用于通信协议的正确性验证. 但由于该方法需要遍历系统所有状态空间, 会面临著名的状态空间爆炸问题, 导致其可验证的系统规模受限. 而时间自动机本身行为的复杂性, 更是加剧了这一问题, 使得通过此方法可验证的协议规模通常仅含有几个节点. 事实上, WSN 系统通常具有较大规模, 包含几十甚至上百个节点. 因此, 传统的模型检验技术无法验证真实规模的 WSN 协议的工作情况. 此外, 模型检验技术仅能处理功能性验证问题, 例如系统能否实现某个目标, 但是无法回答如可以多快实现此目标之类的性能分析问题.

近年来, 基于统计的模型检验技术 (statistical model checking, SMC)[7,8] 被提出, 并得到了广泛关注. SMC 方法避免穷尽搜索系统的所有状态, 而是用基于仿真及统计的方法来处理大规模系统. 其基本工作原理是随机产生足够的系统运行路径的样本空间, 对每一个独立运行判定其是否满足给定的系统规约, 然后用基于统计的方法分析系统是否满足给定规约, 并给出满足该规约的置信概率区间. 与传统的模型检验相比, 其在时间和空间上的消耗大大降低 [9].

基于以上技术, 本文提出了一种系统地分析和评估 WSN 协议的新途径. 在建模方面, 我们首先构造协议的时间自动机模型, 以描述协议在理想环境下的整体工作流程. 然后, 为描述其在真实场景下的活动, 在模型中引入加权分枝对该时间自动机进行扩展, 构建系统的随机时间自动机, 实现模型状态间的随机转换, 以对环境中的不确定因素建模. 例如, 在随机时间自动机中, 系统中的广播信息可以以一定的概率丢失, 而不是理想模型下所有相邻节点均能够成功接收到广播信息. 此外, 随机时间自动机中的节点也会以一定概率进入失效状态, 类似地, 节点失效一段时间后也会以一定概率复活, 重新加入到工作网络中.

在验证阶段, 用经典模型检验技术在理想时间自动机模型上进行协议逻辑正确性的验证, 如验证协议在功能上是否符合其设计目标等. 而在随机时间自动机上, 则可以用基于统计的模型检验来分析和评估系统在真实复杂环境下的性能. 例如当系统中存在信息丢失和节点失效的情况下, 系统仍然能完成既定目标的概率等. 由于基于统计的模型检验技术的复杂度远低于经典模型检验技术, 因此, 对大规模 WSN 系统的性能进行分析成为了可能. 在此基础上, 利用基于统计的模型检验技术, 我们还可

以在协议设计和选择阶段, 实现参数配置, 性能比较等目标.

为系统展现上述途径的可用性及相关技术细节, 本文用该途径对 TPSN (timing-sync protocols for sensor networks)[10] 和 FTSP(flooding time synchronization protocols)[11] 两种著名 WSN 时间同步协议进行了建模和分析. 通过对这两种协议建模和验证发现, 本文所提途径可以有效分析验证大规模 WSN 系统, 并对被检验协议从功能性验证到性能评估等多个角度给出更加完整、充分的解析.

2 背景介绍

2.1 无线传感网络及其协议

无线传感网络 (WSN) 可被嵌入于环境或移动设备中以观察和监测物理世界, 现在已被广泛应用于医疗、环境保护、军事、火山监测等各个方面 [12]. WSN 通常由分布在大范围区域内的大量传感节点协同工作. WSN 系统具有工作环境恶劣、信息无线传输、节点分布广泛等特点. WSN 中的节点能源受限且不便于维护, 应用于其中的协议也与传统网络中的协议差别较大. 因此, 在设计和选择 WSN 协议时应考虑能源受限、硬件受限、网络连接不稳定、传感节点与现实环境结合紧密等一系列问题.

TPSN[10] 和 FTSP[11] 是 WSN 中广泛应用的两种时间同步协议. TPSN 的主要作用是为 WSN 提供全局范围内的时间同步. 其工作流程大致如下 [10]: 首先, 在网络范围内生成一个分层结构. 分层结构以指定节点作为根节点, 其他节点均被分配唯一的层次号, 该阶段被称为分层阶段. 其次, 在生成的分层结构的各分枝上进行节点之间的两两同步, 最终使网络中的所有节点均与根节点达到同步, 该阶段被称为同步阶段. 此外, TPSN 中还包含一些特殊场景, 如新节点加入, 或网络中的节点在同步过程中失效等.

FTSP 是另一种提供 WSN 全局同步的时间同步协议 [11]. 协议开始工作后, 首先用动态根节点选择算法选出唯一根节点, 并将该根节点的时间视为网络中的全局时间, 其他节点最终与根节点同步. 根节点会定期向相邻节点广播发送时间信息作为参考信息. 其他节点接收到足够多的参考信息后, 根据信息中包含的时间戳估算网络的当前全局时间, 调整本地时钟与根节点同步. 节点同步后会转发其接收到的最新时间信息给相邻节点. 此过程持续进行, 直到最终系统中的所有节点均与根节点同步.

2.2 时间自动机及基于统计的模型检验

时间自动机 [3,4] 被广泛应用于实时系统的建模和分析. 单个时间自动机是一个有限状态系统. 系统中包含一系列用于描述系统时间参数的时钟变量. 时间自动机中所有时钟变量的初始值都为 0, 且在系统运行阶段以同样的速率增长. 在时间自动机中, 事件表示为节点间的迁移. 时钟变量以及迁移上的约束条件用于限制时间自动机中的迁移. 节点迁移只有在边上的所有约束条件都满足时才会发生. 时钟约束还可以以不变式的形式约束系统中的节点, 只有不变式的值为真时, 系统才可以进入或停留在该节点. 时间自动机可以用带权的概率分枝实现节点的随机迁移, 扩展为随机时间自动机 [5].

时间自动机理论可被用于验证实时系统的正确性 [3]. 人们通常关注系统两方面的性质, 活性和安全性. 而时间自动机验证方面, 目前主要关注于系统安全性方面性质验证. 系统的安全性验证可通过可达性分析来实现. 在时间自动机上, 可达性分析问题可判定, 一般可以通过构造域模型、差分矩阵等方法计算, 但是复杂度较高, 可验证问题规模比较有限.

近期, 基于统计的模型检验 (SMC)[7,8] 被提出, 来对系统满足相关性质的概率进行分析. SMC 技术用基于随机仿真或运行的方法得到系统的运行样本, 针对每个独立样本判定其是否符合给定的性质.

在此基础上, SMC 通过统计性算法, 如假设检验等, 统计已有样本中运行情况来估算整个模型满足给定性质的概率. 与传统的模型检验技术不同, SMC 用基于仿真的方法产生系统运行路径的样本, 从而避免了穷尽遍历系统所有状态空间所带来的状态空间爆炸问题, 使系统的可验证规模大大增加. 与复杂度极高的经典模型检验技术相比, SMC 技术复杂度较低, 操作性强, 因此可用来对大规模系统进行分析.

2.3 WSN 协议建模与验证相关工作

在文献 [13] 中, Huang 等人给出了描述 TPSN 理想环境下工作流程的时间自动机. 并基于此时间自动机验证了 TPSN 的一系列基本性质, 包括网络中的所有节点是否能与根节点同步, 网络中任一节点与根节点的时钟偏移是否在合理的范围区间内等. 为了更好地描述 TPSN 中的时间, 文章用整数时钟模拟各节点的局部时钟, 以支持系统中时间方面的复杂运算. 但是, 这种处理时间的方法同时也大大增加了模型的复杂性, 使他们只验证了包含 3~5 个节点的系统. 此外, 模型中没有考虑 WSN 环境中存在的信息丢失和节点动态性等不确定因素.

文献 [14~16] 用形式化方法建模和验证了一种广泛应用的 WSN 时间同步协议 FTSP. 在文献 [14] 中, Kusy 等人在用 Promela 语言对 FTSP 建模时考虑了 WSN 系统工作环境的复杂性, 提出系统中的信息会以一定的概率被丢弃. 但由于状态空间爆炸问题, 他们只成功使用工具 SPIN 验证了包含两个节点的系统模型, 且模型中没有考虑节点和链路失效问题. 文献 [15] 中使用 CSP 语言对 FTSP 建模, 并验证了含 2~7 个节点的 FTSP 的多条基本性质, 指出其在理想环境下可以成功选出根节点和实现全局范围的时间同步. 但与文献 [14] 类似, 该文在建模时也没有考虑节点和链路失效等特殊场景. 文献 [16] 中, Tan 等人用时间自动机建模和验证了包含 5 个节点的系统. 在设定的特殊场景下, 发现了 FTSP 中的一个反例. 同样, 他们的模型中没有引入对时钟偏移和链路失效等场景的建模.

在文献 [17,18] 中, Vaandrager 等人采用相似的方法对一种 WSN 协议进行了建模与验证. 他们验证了系统在网络中所有节点均可相互通信的拓扑结构下协议的运行情况, 并分析了模型检验工具所给出的系统反例. 然而, 在他们的模型中, 依然没有考虑通信延迟、通信不稳定等非确定性因素.

在文献 [19] 中, Abo 等人概率模型检验技术及相关工具 PRISM[20] 建模和分析了一个移动 WSN 系统的性能. 文章提出用随机 π- 算子对系统建模, 并用 PRISM 验证系统性能. 由于大多数 WSN 协议是时间相关的, π- 算子对时间的建模粒度差于时间自动机. 而且, 概率模型检验依赖于数值计算, 复杂度高于基于统计的统计模型检验, 因此限制了可验证系统的规模.

综上所述, 现有的对 WSN 协议建模和验证的工作并没有考虑 WSN 协议的特殊性, 而是用与其他协议相同的方法对其建模和分析. 建模方面, 首先, 建模语言不统一, Promela、CSP 等语言可以有效描述高层逻辑行为但是无法描述系统实时行为细节. 而 WSN 协议与时间密切相关, 因此 Promela 等语言所建立的模型不能准确反应相应粒度系统实时行为. 其次, 对 WSN 协议验证时, 多数研究者并没有考虑 WSN 工作环境的特殊性, 因此建模和验证时没有考虑 WSN 中广泛存在的非确定性因素. 在验证分析方面, 首先, 目前对 WSN 协议的验证多是集中于功能正确性方面, 多数工作并没有涉及协议的性能. 其次, 目前用于分析和验证 WSN 协议的模型检验和概率模型检验技术复杂性高, 由于状态空间爆炸问题使得可验证的系统规模受限, 处理的模型系统与现实系统规模差距较大, 参考作用颇为受限. 针对以上问题, 我们在文献 [21] 初步工作的基础上提出了一个面向 WSN 协议的整体建模、分析途径. 首先, 选定时间自动机为 WSN 协议建模语言, 并进行随机迁移扩充以支持对 WSN 系统随机行为的表达. 然后, 引入新的分析技术以扩大可验证系统的规模, 并在功能验证之外进行系统性能分析,

来对 WSN 协议进行全面、整体的分析与评估.

3 基于随机时间自动机的无线传感网络协议建模与分析途径

WSN 协议内容多样, 各有侧重. 但由于 WSN 环境的特殊性, 这些协议中也存在一系列共性, 如信息交换过程中的冗余机制, 时间同步协议中对时间的计算机制等. 基于此, 在对 WSN 协议进行建模和验证时, 本文将提取相关共性, 来采用统一的途径指导对协议框架的建模并提取要验证性质. 在此基础上, 再根据协议本身的特点对模型和待验证性质进行填充和调整.

首先, 在建模语言方面, 由于 WSN 协议行为与时间密切相关, 我们采用时间自动机对其建模. 大部分的 WSN 协议中, 所有传感节点在网络中的作用是相同的. 因此可以建立统一的模板描述系统中所有节点的行为. 然后, 通过给定节点号标识网络中的各节点将 WSN 系统建模为时间自动机网络. 节点之间可以通过同步信息进行交互. 网络拓扑结构可以通过节点间连接关系函数进行配置与描述.

建模与分析方法方面, 首先构建描述协议整体工作流程的时间自动机模型, 描述系统在理想环境下的工作行为. 为了对现实环境中普遍存在的不确定因素进行建模与表达, 用加权分枝实现模型中状态的随机迁移从而得到协议的随机时间自动机模型. 在分析和验证相关协议时, 我们建议采用经典模型检验技术在小规模时间自动机上验证协议在理想环境下的行为, 来检验协议逻辑设计的正确性. 另一方面, 为了分析和评估系统在复杂环境下的工作情况, 可用基于统计的模型检验技术对大规模随机模型进行分析, 以观察协议在不同环境条件、不同网络规模、不同参数取值等条件下的具体性能.

3.1 WSN 协议建模

3.1.1 工作流程时间自动机建模

相较一般通信协议, WSN 协议的工作流程要复杂得多. 因此, 用旧有的将系统所有工作状态进行罗列之后, 再对所有状态两两之间的跳转关系进行链接的方式难以应对大规模系统. 本文提出, 对协议建模时, 在分析阶段, 可采用自顶向下的途径, 根据网络中各节点所处的工作状态或其在当前工作阶段中扮演的角色, 将协议划分为不同的阶段或模块. 然后采用自底向上的途径分别建立各阶段或模块的时间自动机模型, 最后将进行拼接, 整合成描述协议整个工作流程的时间自动机. 这也符合了经典的分而治之的理念, 在面对某单独子流程时, 其复杂度可以有效控制并清晰理解, 从而不会很快的迷失在细节当中.

另一方面, 对 WSN 协议中普遍存在的行为, 我们可以采用统一的途径对其建模, 如:

• 信息发送和接收: WSN 网络中存在大量的信息交互, 如数据共享和同步等. 时间自动机可以清晰地对信息的发送和接收进行建模, 图 1(a) 和 (b) 分别给出了对信息发送和接收建模的简单示例.

• 时间: WSN 协议中存在大量的针对时间的约束和表达. 时间自动机可以通过在节点和迁移上添加时间约束方便地描述协议中的系统反应和信息延迟等时间约束. 图 1(c) 给出了相关简单示例.

3.1.2 复杂环境与动态行为建模

WSN 系统通常部署工作在复杂环境中, 网络中普遍存在大量不确定性因素, 而现有工作所使用的建模语言都未对此现象进行有效表达. 为了在系统模型中描述这些不确定性因素, 可以用加权分枝表达状态的随机迁移来扩展协议的时间自动机. 对时间自动机的扩展分为两步: 首先, 在广播信息的接收

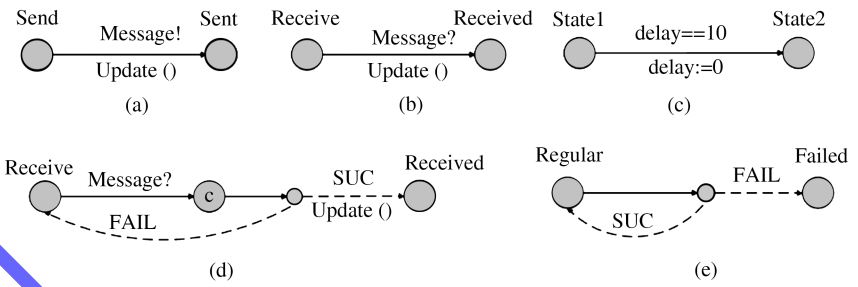


图 1 WSN 协议时间自动机建模示例

Figure 1 Example of modeling WSN protocols by timed automata. (a) Message sending; (b) message receiving; (c) time constraint; (d) weighted branch with message loss; (e) weighted branch with node dynamics

端添加带概率权值的分枝, 使得部分接收节点能成功接收到该信息, 而其余节点不能成功接收到该信息; 然后, 引入新的状态和分枝来表示网络中节点的失效和复活.

存在信息丢失的情况下, 不能保证所有的广播信息均被成功接收. 节点在接收信息时都存在一定的失败概率. 即当一个广播信息被发送时, 部分节点能够成功接收到该信息, 而部分接收节点会在接收信息时失败. 为了模拟该行为, 在时间自动机中引入带概率权值的分枝. 将节点能成功接收信息的分枝权值标记为 SUC, 不能成功接收信息的分枝权值标记为 FAIL. 当标记为 SUC 的边被触发时, 信息被成功接收, 系统根据接收到的信息更新相关参数, 并进入下一状态. 而当标记为 FAIL 的边被执行时, 系统保存在原来状态, 拒绝接收该信息. 图 1(d) 给出了消息 message 的接收端示例.

另一方面 WSN 中的节点能量有限, 且通常工作在恶劣的环境中, 网络中的传感节点会频繁地失效或复活. 为了描述节点的动态性, 系统中引入了一个新的状态 Failed, 表示节点进入失效状态, 并将忽略接下来系统中的所有信息. 理论上, 系统中的节点可以在任意时间失效或复活. 为了描述失效时间的任意性, 处于模型中任意状态的节点均可以以一定的概率进入 Failed 状态. 处于 Failed 状态的节点也可以以一定概率进入系统的初始状态重新加入网络, 表示节点复活. 如图 1(e) 所示, 处于正常状态的节点可以以概率 FAIL 进入失效状态 Failed, 或以概率 SUC 停留在原状态继续工作.

3.2 WSN 协议分析与评估

3.2.1 用传统模型检验进行正确性验证

现有工作对一般通信协议的验证主要集中于用形式化方法验证其功能正确性. 与之类似, 验证 WSN 协议正确性时, 我们采用模型检验技术验证描述协议理想环境下整体工作流程的时间自动机. 在理想时间自动机上, 我们可以检验协议的共有性质, 如协议中是否存在死锁, 以及功能性性质, 如时间同步协议中的所有节点是否能够成功同步等. 通过模型检验, 我们可以验证协议设计逻辑的正确性, 发现协议中可能存在的错误.

虽然经典模型检验技术仅能应对小规模系统, 但其验证结果仍然具有重要的指导意义. 如果能在小规模系统原型上成功验证相关性质, 设计人员对系统的信任度将会得到有效保障. 另一方面, 如果能在验证中发现问题, 那么模型检验技术所提供的反例能有效帮助设计人员进行设计修正, 从而为后期开发节省大量成本. 因此, 其仍然是系统验证中不可或缺的重要一环.

3.2.2 用基于统计的模型检验进行性能评估

与传统模型检验技术相比, 基于统计的模型检验技术复杂性大大降低, 可用于分析大规模网络. 用

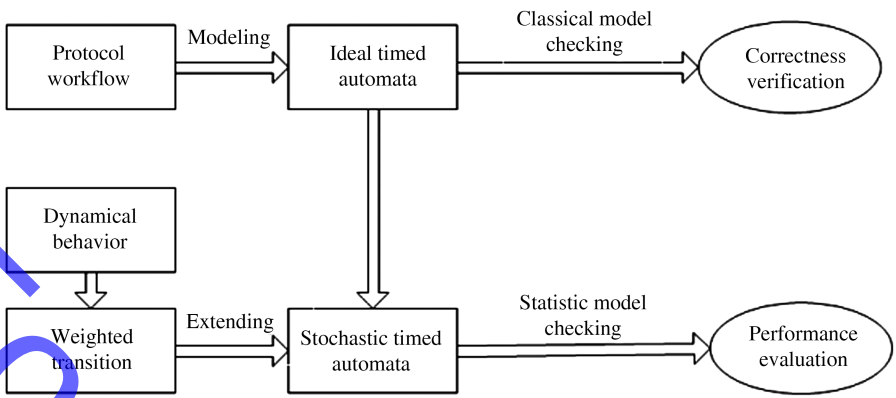


图 2　WSN 协议建模和验证的流程示意图

Figure 2　Workflow of the modeling and analysis of WSN protocol

基于统计的模型检验可以评估协议在不同环境或参数配置下的性能, 如判定协议在一定时间内满足给定性质的概率. 通过配置真实环境的具体概率参数, 可以得出待检验协议在给定环境下的性能, 从而帮助设计人员判断相关协议是否适合工作于特定的目标工作环境. 此外, 协议脚本中存在着很多的可调参数, 这些参数的具体取值对系统性能至关重要. 通过基于统计的模型检验, 我们可以分析、比较同一个参数在不同取值情况下系统满足特定目标性质的概率. 从而达到配置参数的目的.

至此, 我们介绍了对 WSN 协议建模和验证的完整途径, 其总体流程如图 2 所示. 为说明该途径的可用性及其细节, 接下来将给出用该途径对 TPSN[10] 和 FTSP[11] 两个 WSN 协议建模、分析的具体示例.

4　实例研究: 基于随机时间自动机的 WSN 协议建模

4.1　TPSN 建模

TPSN 提供全局范围内的时间同步, 其工作流程分为分层和同步两个阶段 [10]. 分层阶段在网络范围内生成一个分层结构, 由指定节点作为分层结构的根节点, 其余节点则被指定一个层次号. 同步阶段中分层结构各分枝上的节点进行两两之间的信息交换和时间同步, 最终所有节点均与根节点同步. 下面我们将采用自底向上的方法对这两个阶段分别建模.

分层阶段: 分层阶段的时间自动机模型如图 3(a) 所示. 网络进入工作状态后, 指定的根节点首先被标记为 0 层, 并广播分层信息 level_discovery. 其相邻节点接收到该分层信号后, 进入 Discovered 状态并更新自己的层号. 处于 Discovered 状态的节点再广播 level_discovery 信息给其相邻节点, 最终网络中的所有结点均分配到唯一的层号, 加入到分层结构中.

同步阶段: 分层阶段结束后, 系统即进入到同步阶段. 首先, 根节点广播同步信号 time_sync 并直接进入已同步状态 Synchronized. 处于第 1 层的节点接收到该同步信号后, 等待随机时间, 然后开始与根节点进行信息交换. 接收到根节点返回的确认信息后, 第 1 层节点根据根节点时钟调整自己的本地时间, 与根节点同步完成. 第 2 层的节点在监听到上述信息交换后, 等待随机时间, 并开始与第 1 层节点的信息交换和同步. 节点之间的两两同步重复进行, 直到网络中的所有节点均进入已同步状态 Synchronized. 此外, 由于存在时钟偏移, 已同步网络中各节点之间的时间差会随着时间的延续而增大.

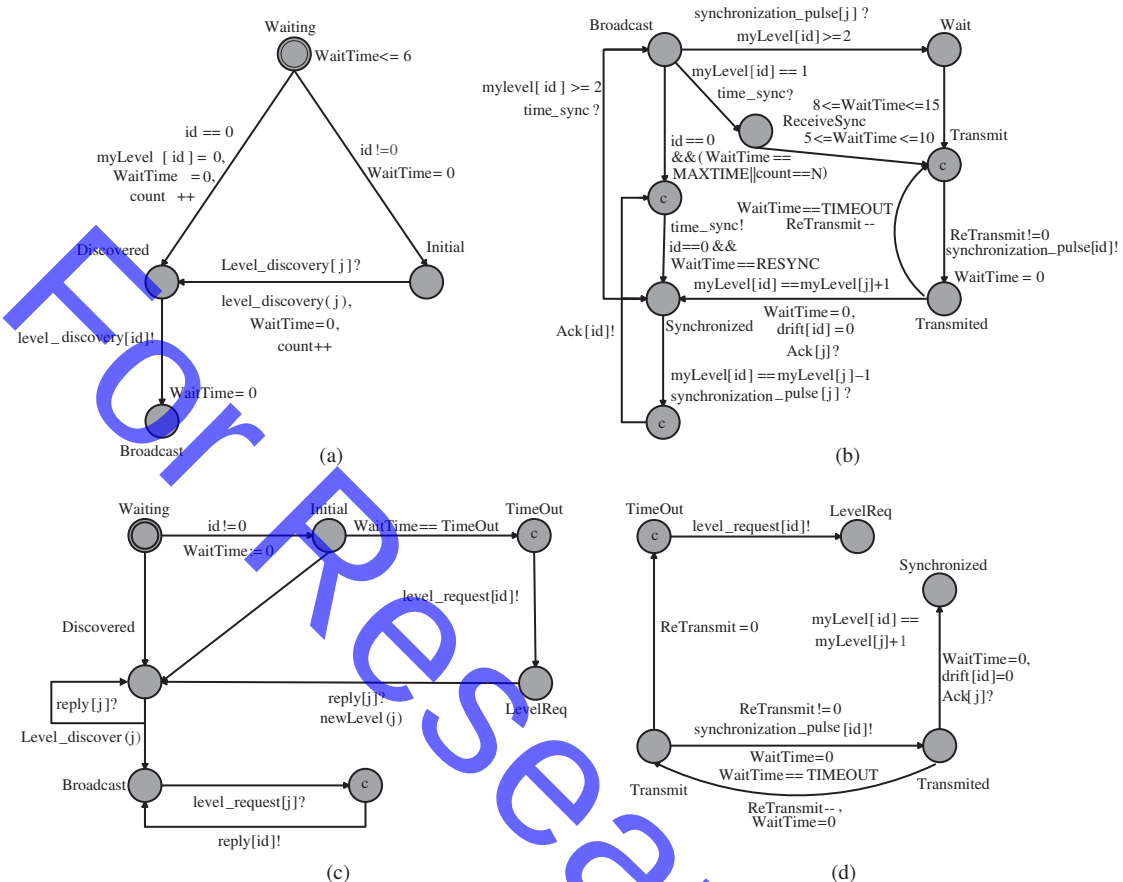


图 3 TPSN 各阶段的时间自动机模型

Figure 3 Timed automata for different phases of TPSN. (a) Level discovery phase; (b) synchronization phase; (c) local level discovery phase; (d) level re-discovery phase

因此, 根节点会定期发送同步信息 time_sync 进行重同步. 同步阶段的时间自动机模型如图 3(b) 中所示.

特殊场景: TPSN 脚本中还提到了协议工作过程中可能遇到的特殊场景, 即新节点加入而引起的局部分层阶段和相关节点失效引起的重新分层阶段. 这两部分的对应模型分别如图 3(c) 和 (d) 所示.

最后, 可将 TPSN 各阶段时间自动机连接起来, 得到图 4 所示的描述其理想环境下整体工作流程的模型. 接下来, 我们对该模型进行概率扩展, 描述 TPSN 在实际环境下的工作场景. 扩展过程中, 主要通过引入加权分枝实现状态的随机迁移, 以描述系统中存在的信息丢失、节点失效等随机行为.

图 5(a) 给出了 level_discovery[id] 信息接收端的例子. 为了描述网络中信息丢失的随机性, 我们在模型中所有广播信息的接收端都增加了类似的随机行为. 同样, 为了模拟网络中传感节点的失效, 模型中引入了新的状态 Failed, 模型中任何常规状态节点均可以随机进入 Failed 状态, 如图 5(b) 所示; 处于 Failed 状态的节点也可以以一定的概率再次进入模型的初始状态, 表示节点复活并重新加入网络中.

将上述 TPSN 的理想模型和随机扩展部分结合起来, 便可得到如图 5(c) 所示的描述 TPSN 整体流程及工作环境中不确定因素的随机时间自动机. 由于篇幅限制, 模型中有些细节并未给出, 读者可访问 http://seg.nju.edu.cn/people/ bl/exp/TPSN.rar 察看 TPSN 的完整模型.

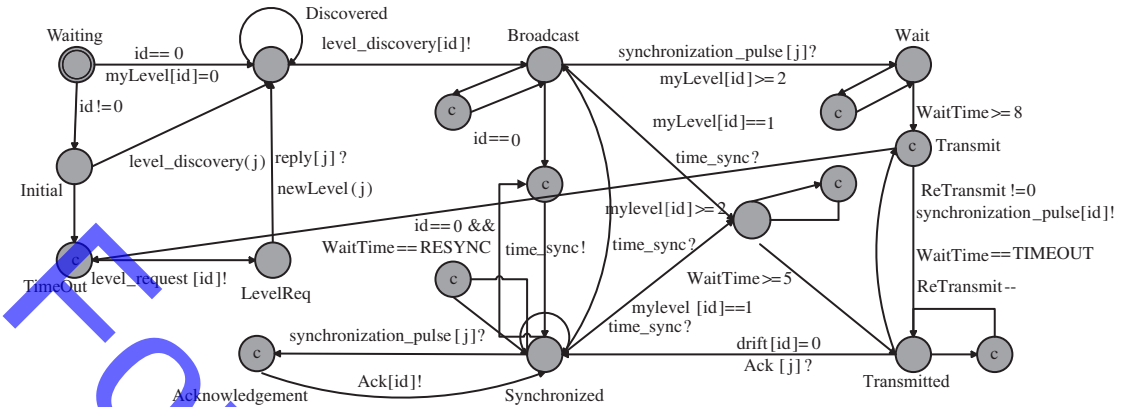


图 4 TPSN 理想时间自动机模型

Figure 4 Ideal timed automata for TPSN

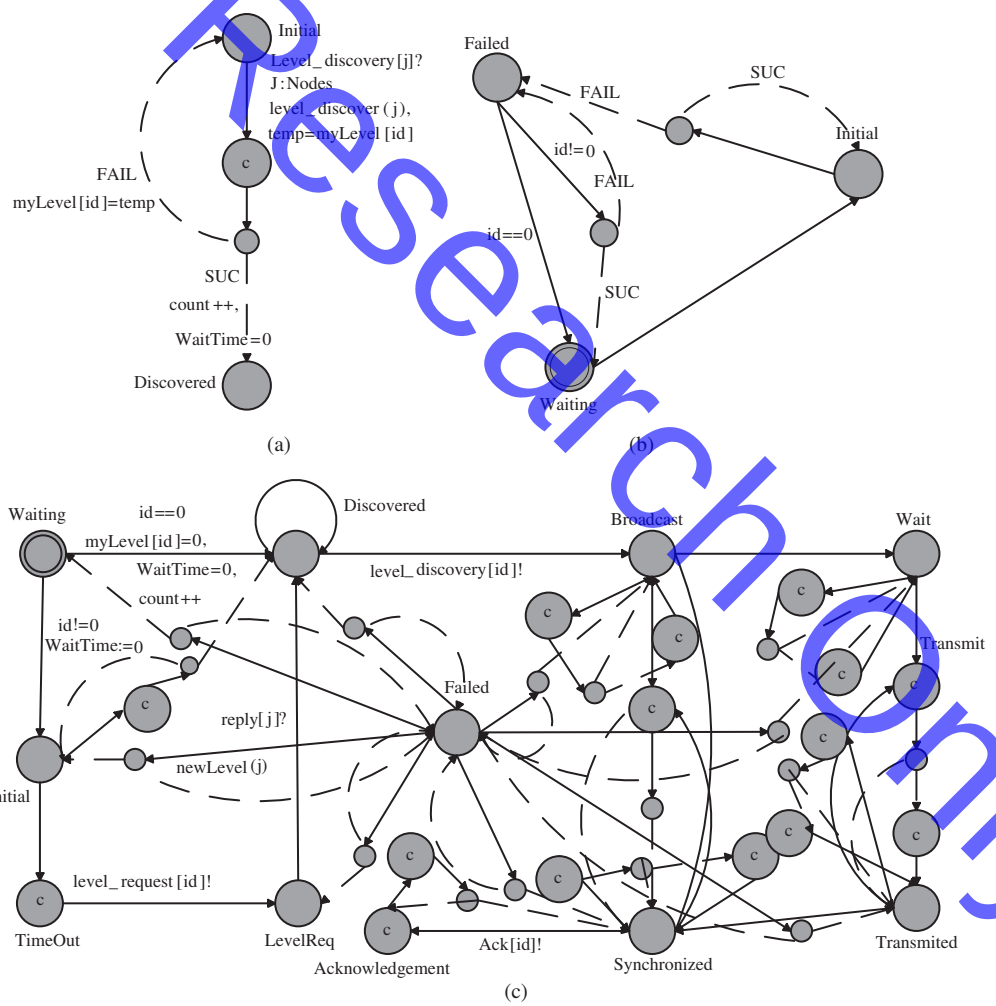


图 5 TPSN 随机时间自动机模型

Figure 5 Stochastic timed automata for TPSN. (a) Example of message loss; (b) example of node dynamic; (c) statistical timed automata for TPSN

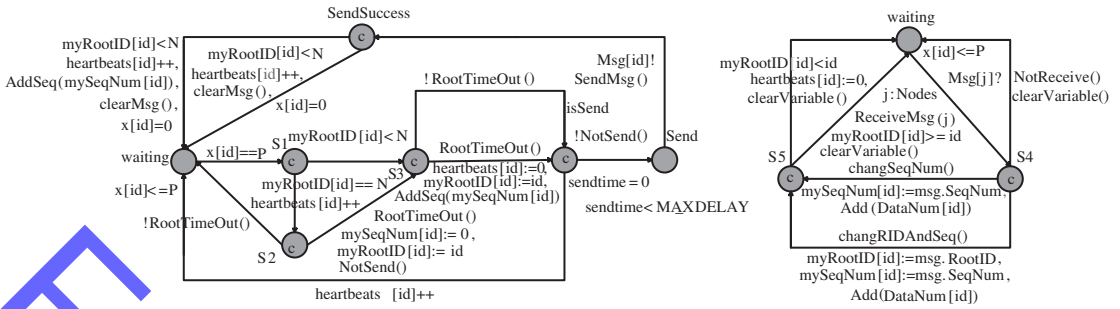


图 6 FTSP 各阶段的时间自动机模型

Figure 6 Timed automata for different phases of FTSP. (a) Sending phase; (b) receiving phase

4.2 FTSP 建模

FTSP 是另一种提供网络全局范围内时间同步的协议, 其工作流程为 [11]: 首先执行根节点选择算法选择合适的节点作为网络的根节点. 然后, 以根节点的时间作为全局时间, 经过多次信息交换, 调整网络中其他所有节点的时钟与根节点同步.

所有节点在 FTSP 协议工作过程中均要与其他节点进行信息交换, 即接收其他节点广播的带时间戳的信息 (参考信息), 并在自身达到同步后, 向相邻节点发送参考信息. 因此, 在协议的整个工作过程中, 节点将交替以信息发送者和接收者的角色工作. 基于此, 我们分别对节点的信息发送和接收行为建模, 最后组合为单个节点的所有行为. 建模时, 假设网络中的所有节点都有唯一的编号 id 作为其标识信息. 协议正确工作时, 设定各节点将选择 id 最小, 即 id=0 的节点作为网络的根节点. 网络初始化时, 各节点当前根节点 myRootID 设为最大值 MAX.

发送阶段行为建模: FTSP 中, 所有节点只有在达到同步后才能向其他节点发送信息. 节点同步分为两种情况 [16]: 1. 节点接收到足够的参考节点, 能成功估算出网络的全局时间, 与根节点完成同步. 2. 节点是当前根节点 (myRootID ==id). 这又分为两种情况: 节点确实是网络全局范围内的根节点或节点在长时间内未接收到其他节点发送的信息, 时钟超时, 宣称自己是根节点. 对以上两种情况进行建模后, FTSP 节点处于发送端行为时的时间自动机模型如图 6(a) 所示.

接收阶段行为建模: 相邻节点发送信息时, 只有当信息发送者的根节点 id 小于或者等于本节点的根节点 id 时, 节点才会接收此条信息. 接收到新的信息后, 节点便会根据信息包含的内容更新其相应参数. 信息接收过程的时间自动机如图 6(b) 所示.

类似地, 通过将上述发送阶段和接收阶段的时间自动机进行整合, 我们可得到图 7(a) 所示的描述 FTSP 完整工作流程的时间自动机. 与 TPSN 类似, 我们在理想时间自动机的基础上加入随机加权分枝以对信息丢失和节点动态等不确定因素建模, 可得到如图 7(b) 所示的 FTSP 的随机时间自动机. 同样, 读者可以访问 http://seg.nju.edu.cn/people/ bl/exp/FTSP.rar 察看该模型的所有细节.

5 实例研究: 基于统计模型检验技术的 WSN 协议分析与评估

本节中我们将在上一节给出的两种自动机的基础上, 对两种协议进行分析与验证. 验证所采用系统为一个包括 N 个节点的模型. 如无特别说明, 所验证网络均为单跳网络. 验证中所用计算机配置如下: Intel(R) Core(TM) 2 Quad Q9500 处理器, 2G RAM , 操作系统为 Windows 7 专业版. 验证所使用工具为时间自动机模型检验工具 UPPAAL(4.1.7 版本)[22]. 统计模型检验实验时伪阴性 (α) 和伪阳性

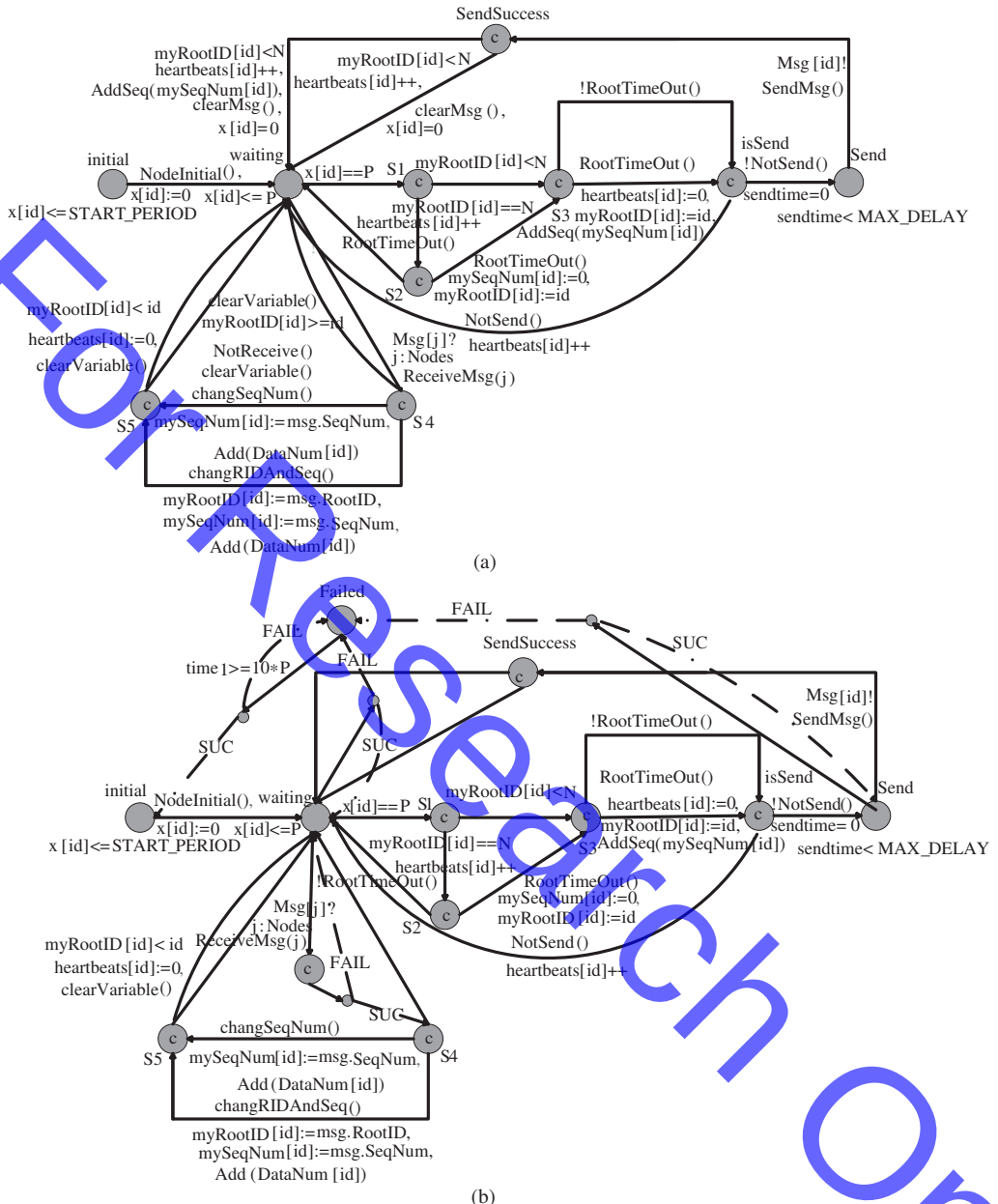


图 7　FTSP 的时间自动机完整模型

Figure 7　Complete timed automata for FTSP. (a) Ideal timed automata for FTSP; (b) stochastic timed automata for TPSN

(β) 等系统参数取值均设为 0.05. 此外, 随机模型中的所使用的失效概率指系统信息丢失或节点失效的概率, 其取值由 FAIL/(FAIL+SUC) 决定.

5.1　TPSN 评估

5.1.1　正确性验证

TPSN 的最终目的是达到全局范围内的时间同步. 协议中, 节点的工作分为分层和同步两个阶段.

表 1 TPSN 正确性验证结果

Table 1 Correctness verification of TPSN

N	3	4	5	6
Level discovery($C(T/M)$)	Y(0.031/13)	Y(0.032/19)	Y(0.25/32)	Y(3.791/17)
Node synchronization($C(T/M)$)	Y(0.031/14)	Y(0.53/20)	Y(43.384/31)	Y(16984.546/47)
Deadlock-free($C(T/M)$)	Y(0.047/15)	Y(0.375/18)	Y(15.865/47)	Y(3603.202/580)

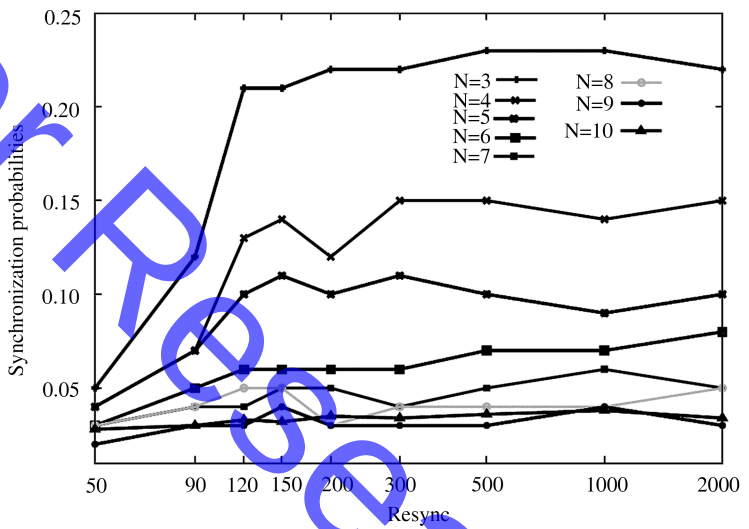


图 8 重同步周期对 TPSN 同步成功概率的影响

Figure 8 RESYNC vs synchronization probability in TPSN

因此, 在验证时可分别验证两个阶段功能实现的正确性. 另一方面, 通信协议中是否存在死锁是一个普遍关注的问题, 因此我们也将对此问题进行验证. 实验中所验证的相关性质采用 CTL 表示如下:

层次分配 (level discovery): 检验所有节点能否被成功分层: A<> ∀(id:Nodes) myLevel[id]< N.

节点同步 (node synchronization): 检验模型中的所有节点能否进入同步状态: A <> ∀(id:Nodes) tpsn(id).Synchronized.

无死锁 (deadlock-free): 检验协议中是否存在死锁: A[ ] not deadlock.

我们用 UPPAAL 分别在包含 3~6 个节点的网络中验证了以上各性质. 验证结果 (性质是否满足 (C: Y/N), 验证时间 (T: s), 内存消耗 (M: MB)) 如表 1 所示, 以上性质均满足. 对 TPSN 的正确性验证说明, TPSN 的设计逻辑正确, 在理想环境如不存在信息丢失、节点失效等不稳定因素的情况下, 能够正常工作.

5.1.2 性能评估

在设计或选择一个协议时, 除协议设计逻辑是否正确外, 人们还关心协议在特定环境下的性能. TPSN 中, 很多因素与协议的性能密切相关, 如重同步周期、网络规模、失效概率等等, 下面我们将针对以上因素, 分别进行分析.

重同步周期: 重同步周期是 TPSN 脚本中的一个可调参数. 在选择重同步周期 (RESYNC) 的具体取值时应仔细权衡: 一方面, 周期既要足够长, 使得一个同步周期内, 网络中的所有节点均能够达到

表 2　TPSN 系统失效概率为 0 和 0.1 的情况下, M 个周期内同步成功的概率

Table 2　Probability of synchronization success in M cycles of TPSN with failure probability as 0 and 0.1

	Failure probability=0			Failure probability=0.1			
N	M=1	M=2	M=3	M=1	M=10	M=100	M=400
3	[0.84,0.94]	[0.95,1]	[0.95,1]	[0.17,0.27]	[0.31,0.41]	[0.65,0.75]	[0.93,1]
5	[0.88,0.98]	[0.95,1]	[0.95,1]	[0.02,0.12]	[0.09,0.19]	[0.27,0.37]	[0.62,0.72]
7	[0.89,0.99]	[0.95,1]	[0.95,1]	[0,0.10]	[0.02,0.12]	[0.07,0.17]	[0.25,0.35]
9	[0.87,0.97]	[0.95,1]	[0.95,1]	[0,0.06]	[0,0.07]	[0,0.09]	[0.06,0.16]
10	[0.88,0.98]	[0.95,1]	[0.95,1]	[0,0.06]	[0,0.07]	[0,0.07]	[0.02,0.12]
20	[0.85,0.95]	[0.95,1]	[0.95,1]	[0,0.05]	[0,0.05]	[0,0.05]	[0,0.05]

同步状态; 另一方面, 为了使网络中节点之间的时间差保持在合理范围内, 周期又要尽可能短. 为研究重同步周期对 TPSN 性能的影响, 我们进行了一组实验, 检查系统在各种重同步周期赋值情况下在一个同步周期内成功同步的概率. 以失效概率赋值 0.1 为例, 具体结果如图 8 所示.

从图中可以看出, 随着重同步周期取值增大, 所有节点在单个周期内同步成功的概率快速增大; 但当取值接近 150 的时候, 同步成功概率趋于稳定. 从图中可以看到, 即使重同步周期取值足够大, 实验中最好情况下一个周期内所有节点均完成同步的概率也仅在 0.2 附近. 网络中无法使所有节点在一个周期内同步的原因是: 极端情况下, 网络中总有节点会因长时间没有接收到信息而超时进入重新分层阶段, 无法与其他节点同步. 此时, 与其继续增大一个同步周期的长度等待所有节点在同一个周期内同步, 不如选择合适的 RESYNC 取值, 使所有节点再次进入下一轮同步. 从图 8 可以看出, 本次实验中, 重同步周期取值为 150 时, TPSN 性能较好. 因此, 接下来的所有实验中, 其取值均设为 150.

网络规模: WSN 中普遍包含大量节点, 这要求相应协议在大规模网络中具有良好的性能. 我们在 TPSN 的随机时间自动机上失效概率为 0 和 0.1 的情况下分别做了两组实验, 考查 TPSN 在网络规模变化的情况下所有节点在 M 个重同步周期内同步成功的概率, 结果如表 2 所示. 实验中, 我们最多验证了包含 100 个节点的系统, 其性能与网络中只有 20 个节点时的性能相似, 因此, 20~100 之间的数据没有在文中给出. 结果显示, 当环境稳定时, 随着节点数目的增多, 相同时间内所有节点均同步成功的概率较为稳定. 说明在稳定环境下 TPSN 的可扩展性较好. 但当环境不稳定时, 相同规模的网络同步成功的概率急剧减小. 说明 TPSN 性能对环境十分敏感, 环境的轻微变化即会引起其性能的急剧恶化.

失效概率: 上组实验表示环境中存在的信息丢失和节点动态变化等情况可能对 TPSN 的性能有较大影响. 为了验证这个结论, 我们在 TPSN 的随机时间自动机上增做了失效概率分别取 0, 0.1, 0.2, 0.3 时, 包含 3 个节点的网络在 M 个周期内所有节点均同步成功的概率. 实验结果如图 9 所示, TPSN 中所有节点同步成功的概率随失效概率的增加而急剧下降. 当失效概率为 0, 即网络在理想环境中工作时, 10 个周期内, 所有节点均可同步成功. 但当失效概率为 0.1 时, 确保所有节点均成功同步则需近 400 周期. 而当失效概率为 0.3 时, 在相同网络里, 5000 个同步周期仍几乎不可能使所有节点均同步成功. 该实验说明 TPSN 对环境中可能存在的信息丢失和节点失效特别敏感. 即使网络中节点数目非常有限, 当工作在稍不稳定环境中时, 协议也几乎不可能正常工作.

以上对 TPSN 的验证和分析表明, 理论上, TPSN 的设计逻辑正确, 在理想环境下能够正常工作. TPSN 的可扩展性较好, 网络规模增大时, 其性能仍能保持稳定, 不会急剧变化. 但 TPSN 对环境敏

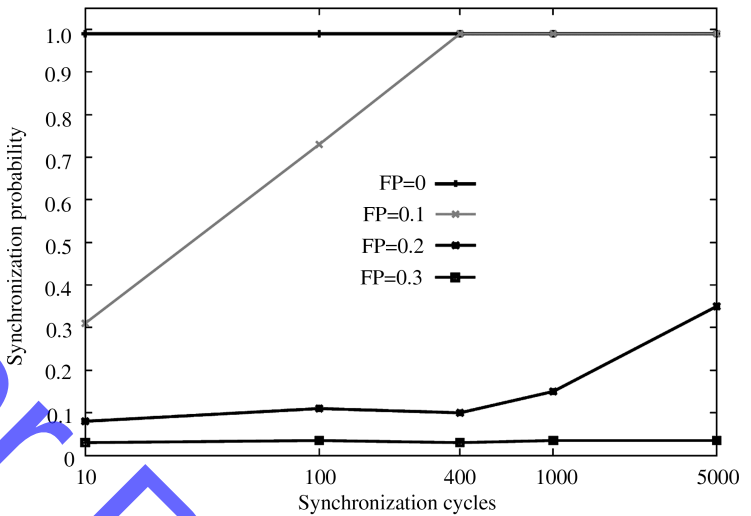


图 9 失效概率对 TPSN 同步成功概率的影响

Figure 9 FAIL vs synchronization probability in TPSN

感, 当工作环境中干扰增加, 特别是信息丢失和节点失效情况加剧时, TPSN 的性能会急剧恶化. 严重时将无法正常工作, 不再能提供全局范围内时间同步.

5.2 FTSP 评估

5.2.1 正确性验证

FTSP 协议可提供全局范围内的时间同步. 协议运行时, 首先利用动态根节点选择算法选举网络中编号最小的节点作为网络的根节点. 然后通过信息交换, 网络中的所有节点与该根节点同步. 因此验证 FTSP 的正确性时, 我们将在其理想时间自动机上对以上性质进行验证. 此外与 TPSN 类似, 我们也将对死锁性质进行检查. 相关性质的 CTL 表示如下:

根节点选择 (root election): 检验能否成功选择 id 最小的节点为根节点: A <> ∀(id:Nodes) myRootID[id] == 0.

节点同步 (node synchronization): 检验除根节点外的其他节点能否接收到足够的参考信息, 进入同步状态: A <> ∀(id:Nodes∧ id!=0) DataNum[id]==VALID_LIMIT.

无死锁 (deadlock free): 检验协议中是否存在死锁: A[] not deadlock.

验证结果 (性质是否满足 (C: Y/N), 验证时间 (T: s), 内存消耗 (M: MB)) 如表 3 所示. 由于状态空间爆炸问题, 无死锁性质只在网络中仅包含两个节点时给出了验证结果, 证明协议中不存在死锁. 而根节点选择及节点同步两条功能性性质, 都被证明并不满足. 模型检验工具给出了相应的反例, 分析发现, 当网络中有两个以上的节点宣称自己为根节点时, 可能在网络中形成一个无限循环, 并将网络分为相对独立的若干个子网络.

5.2.2 性能评估

FTSP 的随机时间自动机模型中, 网络中的所有节点在根节点选择成功后, 即可接收到根节点发送的一系列的参考信息. 节点接收到足够的参考信息后, 将估算系统当时的全局时间, 与根节点同步. 显然, 该协议的重点在于其中的动态根节点选择算法. 因此, 性能评估时, 我们主要考查在不同网络规

表 3　FTSP 正确性验证结果

Table 3　Correctness verification of FTSP

N	Root election($C(T/M)$)	Node synchronization($C(T/M)$)	Deadlock-free($C(T/M)$)
2	N(0.125/17)	N(0.093/22)	Y(78/296)
3	N(0.343/18)	N(0.343/25)	N/A
4	N(0.686/27)	N(0.718/46)	N/A
5	N(0.716/82)	N(2.839/117)	N/A
6	N(4.462/68)	N(13.494/277)	N/A
7	N(12.543/345)	N(67.517/265)	N/A
8	N(38.516/470)	N(344.279/778)	N/A
9	N(3125.159/198)	N/A	N/A

表 4　FTSP 在失效概率为 0 和 0.1 的情况下, T 时间内根节点选择成功的概率

Table 4　Probability of root election success in time T of FTSP with failure probability as 0 and 0.1

	Failure probability=0			Failure probability=0.1				
N	100	130	150	100	200	300	500	1000
3	[0.33,0.43]	[0.73,0.83]	[0.92,1]	[0.18,0.28]	[0.65,0.75]	[0.80,0.90]	[0.93,1]	[0.95,1]
5	[0.29,0.39]	[0.77,0.87]	[0.93,1]	[0.04,0.14]	[0.40,0.50]	[0.60,0.70]	[0.86,0.96]	[0.95,1]
7	[0.35,0.45]	[0.78,0.88]	[0.93,1]	[0,0.09]	[0.26,0.36]	[0.44,0.54]	[0.76,0.86]	[0.92,1]
9	[0.37,0.47]	[0.80,0.90]	[0.94,1]	[0,0.07]	[0.16,0.26]	[0.31,0.41]	[0.64,0.74]	[0.89,0.99]
10	[0.40,0.50]	[0.83,0.93]	[0.94,1]	[0,0.06]	[0.12,0.22]	[0.26,0.36]	[0.55,0.65]	[0.83,0.93]
20	[0.40,0.50]	[0.86,0.96]	[0.95,1]	[0,0.05]	[0,0.06]	[0.01,0.11]	[0.09,0.19]	[0.23,0.33]

模、不同参数配置和不同环境因素下, 根节点选择成功的概率, 即 Pr[time< T](<> ∀(id:Nodes) myRootID[id]==0), 其中 T 表示系统运行时间. 由于建模所采用的 FTSP 协议给出了具体代码实现, 其中对所有参数均给出了具体取值, 因此, 在本实验中我们主要考察了系统中网络规模及失效概率不同情况下 FTSP 的具体性能.

网络规模: 为评估 FTSP 的可扩展性, 我们验证了网络中节点数目 N 不同取值时根节点选择成功的概率, 结果如表 4 所示, 包括失效概率分别设为 0 和 0.1 的两组实验. 结果显示, 失效概率为 0 时, 即网络工作环境相对稳定的情况下, 随着网络规模的增大, FTSP 中根节点选择成功的概率不仅不会减小, 反而有所增大. 这是因为 FTSP 是一种洪泛式时间同步协议, 信息传输时有冗余处理机制, 协议中所有信息均为广播发送. 而网络中的所有节点根节点选择成功后, 均会向其相邻节点广播其当时的根节点及时间信息. 随着网络中节点数目的增多, 网络中包含根节点及时间信息的广播信号也会相应增加. 网络中的节点便可以更快地接受到足够的参考信息, 以选择根节点及估算网络全局时间. 另一方面, 当网络中通信环境不理想, 如信息丢失和节点失效等的失效概率为 0.1 时, 环境中的动态因素也会影响网络中的节点, 使其无法正常通信, 因此, 网络性能会随网络规模的增大而相应降低.

表 4 中的数据表明, FTSP 具有较强的可扩展性, 其洪泛式的广播机制使得节点可以更快接收到正确的参考信息, 导致网络性能在规模增加时反而得到一定提升. 但实际工作环境中, 由于信息丢失、节点失效等情况的存在, FTSP 工作过程中会由于环境的干扰出现一定程度性能降低. 此外, 比较表 2

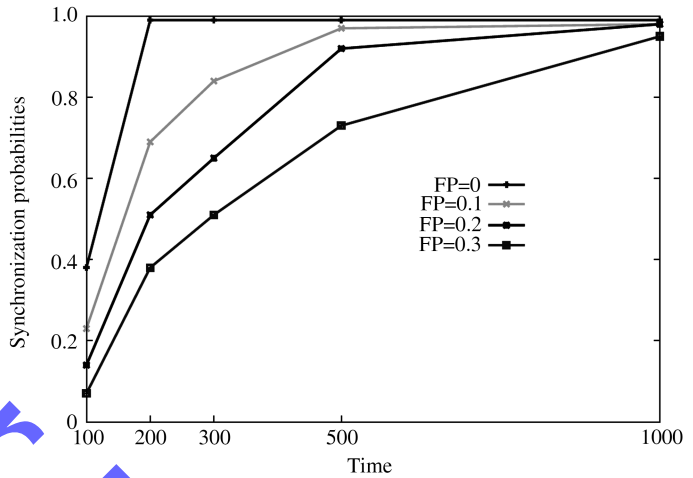


图 10 FTSP 中失效概率对根节点选择的影响

Figure 10 FAIL vs probability of root election success in FTSP

与表 4 中的数据还可以发现, 当环境中存在信息丢失和节点失效时, FTSP 性能下降趋势与 TPSN 相比要平缓得多, 说明其抗干扰能力较强. 为进一步证实此结论, 接下来我们将进行更多的实验.

失效概率: 从表 4 中可以看出, 网络中失效概率变化时, FTSP 协议中根节点选择成功的概率也会随之变化. 为了探究环境对 FTSP 性能的影响, 我们以包含 3 个节点的 FTSP 随机时间自动机为例, 在其上分别做了失效概率 (FP) 为 0, 0.1, 0.2, 0.3 的 4 组实验, 结果如图 10 所示.

从图中可以看出, 随着失效概率的增大, 即网络中信息丢失和节点失效情况的加剧, FTSP 的性能逐渐下降. 但与 TPSN 对失效概率非常敏感相比, FTSP 的性能下降趋势缓和很多, 且随着时间增长仍然可以成功完成任务. 这说明网络中干扰因素的增大会造成 FTSP 中节点在短时间内无法正常完成工作. 但给定足够时间, 由于洪泛机制的影响, 一旦接收到足够的参考信息仍可成功完成相关任务.

以上对 FTSP 的验证和分析表明, FTSP 协议设计过程中存在错误. 在特定情况下, 系统中会出现多个节点宣称为根节点而将网络分为多个子网络, 使网络无法完成全局同步的场景. 性能上, FTSP 抗干扰能力明显优于 TPSN. FTSP 中采用的信息洪泛机制可以帮助协议抵抗干扰, 更快地完成工作任务.

6 总结

本文提出了一种对 WSN 协议建模和评估的通用途径. 首先根据协议的工作流程分阶段、自底向上地建立其理想情况下的时间自动机. 为描述现实中广泛存在的信息丢失和节点失效等不确定性, 可用随机带权迁移扩展相关时间自动机, 得到协议的随机时间模型. 验证阶段, 可用模型检验技术在协议的理想时间自动机上检验协议的功能性性质, 以验证逻辑正确性. 为对协议性能评估和分析, 则可利用基于统计的模型检验技术对其随机时间自动机进行数值分析.

为了说明此途径的可行性及扩展性, 文章以 TPSN 和 FTSP 两种著名的 WSN 时间同步协议为例, 展示了对协议建模和评估的细节. 结果显示模型检验可用于验证协议功能正确性, 并在协议的设计逻辑存在错误的情况下给出反例. 基于统计的模型检验则可用于分析大规模系统 (如包含 100 个节点的网络), 并在对协议的随机时间自动机进行数值分析的基础上, 进行协议参数配置、性能预测及比

较等.

在下一步工作中, 我们将对更多的 WSN 协议采用本文所提框架进行分析与评估, 进一步总结相关共性, 丰富此框架. 同时, 将基于所总结 WSN 协议共性, 在此框架指导下开发面向 WSN 协议的辅助建模、验证、评估工具, 帮助设计人员更加方便的建立模型并自动进行模型检验与性能分析.

参考文献

1 Yick J, Mukherjee B, Ghosal D. Wireless sensor network survey. Comput Netw, 2008, 52: 2292–2330

2 Dwyer M B, Robby J H, Păsăreanu C S, et al. Formal software analysis emerging trends in software model checking. In Proceedings of the 29th International Conference on Software Engineering. Minneapolis, 2007. 120–136

3 Alur R, Dill D L. A theory of timed automata. Theor Comput Sci, 126, 1994. 183–235

4 Bengtsson J, Wang Y. Timed automata: semantics, algorithms and toos. Lect Notes Comput Sci, 2004, 3098: 87–124

5 David A, Larsen K G, Legay A, et al. Time for statistical model checking of real-time systems. Lect Notes Comput Sci, 2011, 6806: 349–355

6 Clarke E, Grumburg O, Peled D. Model Checking. MIT Press, 1999

7 Younes H L S, Simmons R G. Probabilistic verification of discrete event systems using acceptance sampling. Lect Notes Comput Sci, 2002, 2404: 223–235

8 Younes H L S. Verification and planning for stochastic processes with asynchronous events. PhD thesis. Carnegie Mellon, 2005

9 Basu A, Bensalem S, Bozga M, et al. Statistical abstraction and model-checking of large heterogeneous. Int J Softw Tools Tech Trans (STTT), 2012, 14: 53–72

10 Ganeriwal S, Kumar R, Srivastava M B. Timing-sync protocol for sensor networks. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems. Los Angeles: ACM press, 2003. 138–149.

11 Maróti M, Kusy B, Ssimon G, et al. The flooding time synchronization protocol. In: Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems. Baltimore, 2004. 39–49

12 Sundararaman B, Buy U, Kshemkalyani A D. Clock synchronization for wireless sensor networks: a survey. Ad Hoc Netw, 2005, 3: 281–323

13 Huang X, Singh A, Smolka S A. Using integer clocks to verify the timing-Sync sensor networks protocol. In: Proceedings of the 2nd NASA Formal Methods Symposium. Washington, 2010. 77–86

14 Kusy B, Abdelwahed S. FTSP protocol verification using SPIN. Technical Report ISIS-06-704, Institute for Software Integrated Systems, 2006

15 McInnes A I. Model-checking the flooding time synchronization protocol. In: Proceedings of the 7th IEEE International Conference on Control & Automation. Christchurch, 2009. 422–429

16 Tan L, Bu L, Zhao J, et al. Analyziang FTPS robustness with timed automata. In: Proceedings of the 2nd Asia-Pacific Symposium on Internetware. Suzhou, 2010

17 Heidarian F, Schmaltz J, Vaandrager F. Analysis of a clock synchronization protocol for wireless sensor networks. Lect Notes Comput Sci, 2009, 5850: 516–531

18 Schuts M, Zhu F, Heidarian F, et al. Modelling clock synchronization in the chess gMAC WSN protocol. Workshop on Quantitative Formal Methods: Theory and Applications EPTCS, 2009. 1–14

19 Abo R, Barkaoui K. A performability analysis of mobile wireless sensor networks with probabilistic model checking. In: Proceedings of the Wireless Advanced. London, 2011. 283–288

20 PRISM. http://www.prismmodelchecker.org/

21 Zhang F, Bu L, Wang L, et al. Modeling and evaluation of wireless sensor network protocols by stochastic timed automata. In: Proceedings of the 6th International Workshop on Practical Applications of Stochastic Modelling (PASM2012), London. 2012

22 UPPAAL. http://www.uppaal.com/

Modeling and analysis of wireless sensor network protocols by stochastic timed automata and statistical model checking

ZHANG FengLing, BU Lei*, WANG LinZhang, ZHAO JianHua & LI XuanDong

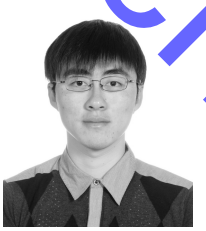
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023*, China*
*E-mail: bulei@nju.edu.cn

Abstract Recently, sensor technology has seen significant achievement. Meanwhile, Wireless Sensor Network (WSN) technique has attracted lots of attention because of its open and dynamic behaviors, and has become a very important component of the Internet-based computing. The behavior of WSN system is very complex and may encounter lots of stochastic uncertainties and disturbances like message loss and node dynamics. Furthermore, WSN system is difficult to change and maintain once it is deployed. Thus, it is critical to ensure the quality of the low level protocols of WSN system in design phase. Designers should ensure the logical correctness of a protocol, as well as evaluate the performance of the protocol under certain target environments.

To handle these issues, in this paper, we propose a framework to analyze and evaluate WSN protocols based on stochastic timed automata and statistical model checking. We propose to address the modeling of the uncertainties in the realistic behaviors by weighted stochastic transitions in the timed automata model of the protocol. Furthermore, we propose that the performance of the protocol under realistic environments should be evaluated by statistical model checking which is much cheaper and more scalable. To illustrate the feasibility and detail of the modeling and verification approach presented in this paper, two well-known WSN protocols, Timing-sync Protocol for Sensor Networks (TPSN) and Flooding Time Synchronization Protocol (FTSP), are studied thoroughly throughout the paper.

Keywords wireless sensor network protocol, timed automata, stochastic timed automata, model checking, statistical model checking

ZHANG FengLing was born in 1987. She received her B.S. degree in software engineering from Nanjing University of Posts and Telecommunications in 2010. She is now a postgraduate student in Department of Computer Science and Technology at Nanjing University. Her research interests include formal methods, protocol verification and software engineering.



BU Lei was born in 1983. He is an Assistant Professor in the Department of Computer Science and Technology at Nanjing University. He received his B.S. and Ph.D. degree in computer science from Nanjing University in 2004 and 2010 respectively. He has been a visiting scholar in Carnegie Mellon University, Fondazione Bruno Kessler, and University of Texas at Dallas. His main research interests include formal method, model checking, especially verification of hybrid system and cyber-physical system.

WANG LinZhang was born in 1973. He is an Associate Professor in the Department of Computer Science and Technology at Nanjing University. He received his Ph.D. from Nanjing University in 2005. His research interests span modeling, analysis, testing and verification in software engineering area, focusing on practical techniques and tools for improving the efficiency of software development, and improving the quality of software under development.

ZHAO JianHua was born in 1971. He is a Professor in the Department of Computer Science and Technology at Nanjing University. He received his Ph.D. from Nanjing University in 1999. His research interests include software engineering, formal method, especially model checking technique for real-time systems. He has directed or anticipated in several National Natural Science Foundation projects and Hi-Tech projects of China.