

无线传感器网络的 NS2 扩展与仿真机制研究

张小庆¹ 李春林² 张恒喜^{2,3}

(黄冈师范学院数学与计算机科学学院 黄冈 438000)¹

(武汉理工大学计算机科学与技术学院 武汉 430063)² (徐州空军学院基础课程部 徐州 221000)³

摘要 分析了 NS2 的工作原理和仿真机制,重点介绍了分裂对象模型和离散事件调度模型。以无线传感器网络作为重点,阐明了 NS2 对无线传感器网络的扩展机制,包括节点能耗模型扩展、移动节点扩展以及相应路由协议扩展。在此基础上,介绍了 NS2 仿真传感器网络路由协议的代码调用过程及仿真模型,并以分簇路由协议 LEACH 为实例进行了仿真实验,从而为解决 NS2 下无线传感器网络协议仿真的代码扩展问题提供了参考。

关键词 NS2, 无线传感器网络, 协议扩展, 仿真, LEACH

中图法分类号 TP393 文献标识码 A

Extension and Simulation in NS2 for Wireless Sensor Networks

ZHANG Xiao-qing¹ LI Chun-lin² ZHANG Heng-xi^{2,3}

(College of Mathematics and Computer Science, Huanggang Normal University, Huanggang 438000, China)¹

(School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430063, China)²

(Department of Fundamental Courses, Xuzhou Air Force Academy, Xuzhou 221000, China)³

Abstract In this paper, the principle and mechanism of NS2 were analyzed. The split-object model and discrete-event dispatch were emphasized, taking sensor network as a key, the extension mechanism of sensor network for NS2 was discussed, including energy consumption model, mobile node extension and routing protocol. Based on these works, the code transferring process and simulation model of wireless sensor networks routing protocol in NS2 were introduced. Then taking LEACH as an example, some simulation experiments were carried out. The network simulation and codes extension in NS2 become easily by the help of such works.

Keywords NS2, Wireless sensor networks, Protocol extension, Simulation, LEACH

最早起步于 20 世纪 90 年代的无线传感器网络作为 21 世纪最有影响的 10 大技术之一,已经成为目前的研究热点,并且越来越受到学术界、军界和工业界的关注。目前的研究主要集中在节能、时间同步、节点定位、通信协议和拓扑控制方面,而常用的仿真平台主要包括 NS2^[1], OPNET^[2], OMNET++^[3], GloMoSim^[4] 等。NS2 作为学术界常用的离散事件网络仿真工具,包括了大量的有线或无线、本地连接或卫星连接进行 TCP 协议、路由算法、多播协议仿真的网络协议、调度器和工具,它主要致力于 OSI 参考模型的仿真,也包括物理层的行为;OPNET 是商业化的通信网络仿真平台,采用网络、节点和过程 3 层模型实现对网络行为的仿真,其无线模型是采用基于流水线的体系结构来确定节点间的连接和传播,用户可指定频率、带宽、功率和包括天线增益模式及地形模型在内的其它特征;OMNET++ 是面向对象的模块化离散事件仿真工具,与 NS2 类似,它主要面向 OSI 模型,可以执行上千节点的仿真,并提供图形化的网络编辑器和网络、数据流查看工具;GloMoSim 是可扩展的用于无线和有线

网络的仿真系统,根据 OSI 进行分层的建模设计。应用层、传输层、网络层、数据链路层、数据包接收模型、无线电模型和无线电波传播模型等不同的层次之间采用标准的 API 进行仿真。

鉴于 NS2 的开源性和使用的广泛性,本文以无线传感器网络为仿真对象分析了 NS2 的工作原理和仿真机制,在节点能耗模型、移动节点扩展以及路由协议上详细剖析了传感器网络的扩展思路。

1 NS2 的仿真机制^[5]

NS2 是基于分裂对象模型的离散事件仿真器,使用 Otcl 和 C++ 两种语言进行对象的互操作,充分发挥了两种语言的优势。

1.1 分裂对象模型

1.1.1 映象创建

NS 的构件库是层次结构的,其中的构件通常由相互关联的两个类来实现,一个在 C++ 中,一个在 Otcl 中。构件的主

到稿日期:2010-09-20 返修日期:2010-12-13 本文受国家自然科学基金项目(60970064, 60773211),教育部新世纪优秀人才计划项目(NCET-08-0806),湖北省杰出青年基金项目(2008CDB335),武汉市科技攻关项目(201010621207),霍英东教育基金项目(121067),中央高校基本科研业务费专项资金(2010-II-003, 2010-ZX-031, 2010-YB-19),软件开发环境国家重点实验室开放课题(SKLSDE-2009KF-2-02)资助。

张小庆(1984—),男,博士生,主要研究方向为无线传感器网络与网格计算, E-mail: zqxqzy@whut.edu.cn; 李春林(1974—),女,教授,博士生导师,主要研究方向为网格计算与高性能计算机网络; 张恒喜(1972—),男,博士生,副教授,主要研究方向为计算机网络与网格计算。

要功能通常在 C++ 中实现, Otel 则主要提供 C++ 面向对象用户配置的接口。在分裂对象模型中, Otel 类称为解释类, 而 C++ 类称为编译类。NS 通过 TclClass 把两种语言的对象和变量联系起来。

图 1 是一个实例化的分裂对象模型及映像对应关系^[6]。

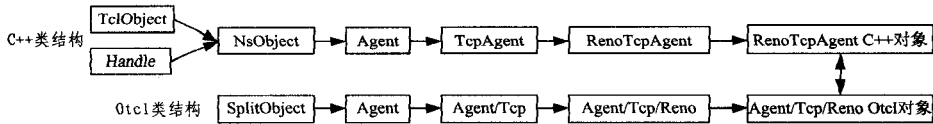


图 1 Otel 类与 C++ 类的对应关系

1.1.2 映像登记

从映像对象的产生过程来看, 在创建解释对象及相应的编译对象之前, 解释对象和编译对象的映像关系应该明确, 而这个过程是通过 TclClass 类的登记功能完成的。图 1 中, 在 RenoTcpAgent 的构造函数中, 将 Otel 类名 Agent/TCP/Reno 作为参数传递给基类 TclClass 的构造函数, 从而确定 Agent/TCP/Reno 与 RenoTcpAgent 的关联关系。

1.1.3 变量绑定

大多数情况下, 获取编译的成员变量只能通过编译代码, 而获取解释的成员变量则只能通过解释代码。但是, 如前所述, C++ 和 Otel 的成员变量能够相互访问, 就需要进行一种双向的变量绑定, 使得解释对象的成员变量和编译对象的成员变量都能够访问相同的数据; 且当任何一边的变量值改变时, 另一边的变量也变为相同值。这种绑定是由编译的构造函数在对象实例化的时候建立的, 它也可以自动地被解释对象以实例变量的身份获得。NS 已经支持 5 种不同的数据类型: 实型 (reals)、带宽变量 (bandwidth valued variables)、时间变量 (time valued variables)、整型 (integers) 和布尔型 (booleans)。如何在 Otel 中说明这些值的语法, 是随类型不同而不同的。

1.1.4 command 方法

要在 Otel 对象中调用对应的 C++ 对象的方法, NS 是通过 command() 函数实现的。对于每个 TclObject, NS 为其 Otel 中的解释对象建立一个实例过程, 即 cmd{}。作为一个挂钩使编译的影像可以执行某些方法。过程 cmd{} 自动调用影像对象的方法 command(), 并将自身的参数作为一个参数数组传递给 command() 方法。用户可以通过以下两种方法来激活 cmd{}: 显示调用过程, 将所要进行的操作作为第一个参数; 隐式调用, 就如同有一个同名的实例对象作为所需的操作。大多数模拟脚本程序都采用隐式调用。

1.2 离散事件调度

NS 是一个事件驱动的模拟器, 它的整个模拟过程是由 Tcl 类的 Simulator 来定义和控制的。Simulator 提供了一套接口, 用于配置一次模拟和为这次模拟选择一个时间调度方案。一个 Tcl 脚本通常以创建这个类的实例为开始, 然后调用各种方法来创建节点、拓扑结构并设置模拟的其他方面。目前, NS 支持 2 种类型的事件调度器: 非实时的 (none real-time) (包括 3 种: 简单链表 (a simple linked-list)、堆 (heap)、时间队列 (calendar queue)) 和实时的 (real-time)。调度器通过选择下一个最早的事件执行, 然后返回继续执行下一个事件的工作方式。调度的时间单位为秒。目前, 模拟器均是单线程的, 所以任意时刻只能执行一个事件。如果同一时刻有多

Otel 对象 Agent/TCP/Reno 与 C++ 对象 RenoTcpAgent 互为映像关系, 其上层的父类也分别对应为映像关系。其中, Otel 类均来自于 SpliObject 类的继承, 而 C++ 类均来自于 TclObject 类的继承, 它们产生的对象分别为解释对象和编译对象, 两种对象互为影像对象。

个事件需要执行, 那么将按照先调度先分配的原则执行。同时发生的事件将不被调度器重新排序, 所有调度器对应同一输入, 应该产生相同的分派顺序。

2 无线传感器网络的 NS2 扩展

NS2 基本按照 OSI 的 7 层模型划分仿真构件库的代码库。每层协议的功能在相应的协议代码中完成, 而层间的接口则是通过 Otel 配置和 C++ 的 Command() 函数联合实现, 它负责分组在协议层间的交付。NS2 仿真传感器网络涉及到很多构件库, 文件扩展主要包括以下源文件和头文件^[7,8]: 应用层 (app. [cc, h])、无线信道 (channel. cc)、逻辑链路层 (ll. h)、MAC 层 (mac. cc, mac-sensor. [cc, h], mac-sensor-timers. [cc, h])、无线物理层 (phy. [cc, h], wireless-phy. [cc, h])、移动节点 (mobilenode. cc)、数据包 (packet. [cc, h])、trace (cmu-trace. [cc, h])。传感节点、agent 和 ll 层的功能定义在了 ns-ranode. tcl, rcagent. [cc, h], rca-ll. [cc, h], resource. [cc, h], energy. [cc, h] 文件中。

2.1 节点扩展

节点是网络拓扑必不可少的重要部分, NS2 支持无线传感器网络的仿真, 但需要对其节点功能进行扩展。

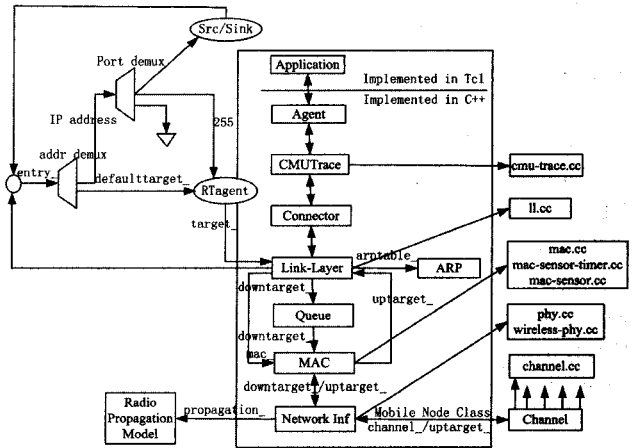


图 2 移动节点模型

图 2 是 NS2 中移动节点的模型。Application 创建数据包发送到 Agent (注: 在 NS2 中, 网络节点间实际没有真实数据传送, 它用到的是一种给定长度的虚拟数据形式, 因此数据包有大小但没有包含任何实际数据), Agent 执行传输层和网络层相关函数 (Agent 相当于执行协议栈中传输层和网络层的功能) 后将数据包发送到 CMUTrace, CMUTrace 又将这个包的统计量写入统计文件, 然后继续将数据报通过连接器发送给链路层进行链路处理。一段短时间的延时后, 数据包从 Link-Layer 到达消息队列。此时如果消息队列有数据包等

待,则数据包进入队列。一旦数据从消息队列中移除被发送至 MAC,就执行介质访问协议。最终,数据包被发送到 Network Interface 层,相应的传输能量在此也被加到数据包中,然后将其发送到设定的物理传输信道 Channel。Channel 发送一个数据包的拷贝到另一节点的传输信道。接收节点按照与发送的相反方向逐层到达应用层。至此,就完成了两个节点间数据包的发送与接收过程。

为实现传感节点的能量自调节功能,NS2 再次对移动节点进行了扩充。图 3 是对图 2 中节点的功能补充,补充的新节点特征是资源(Resource)(包括邻居资源、能量资源)和资源管理(Resource Manager)。Resource Manager 提供了应用层(Application)与单个资源的接口,Resource 可以是任何需要跟踪监测的量,如能量或邻居节点。Application 通过 Resource Manager 利用以下 3 个函数不断更新节点相关资源。add:添加相关资源给节点;remove:移除节点相关资源;query:查询目前节点资源数量。

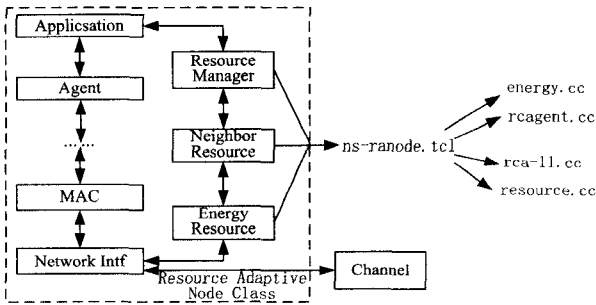


图 3 资源自适应节点模型

资源管理在应用层和每个节点的资源间提供了一个通用的接口,应用层通过这个接口可以动态地获知和更新能源信息。本文关注的能源信息主要是节点的剩余能量,例如在初始仿真时设定了每个节点的初始能量,在一个节点执行数据处理或者数据发送时,应用层将通过这个接口的相关函数减去执行上述操作所消耗的能量。

2.2 能耗扩展

Wireless-phy.cc 文件中扩展了传感器网络的通信能耗模型,称为简单的节点广播能耗模型^[9]。如图 4 所示,模型由功率放大器能耗和射频电路能耗两部分组成,其中发送方能耗等于发送射频电路的能耗加上功率放大器的能耗,接收方的能耗等于接收射频电路能耗。

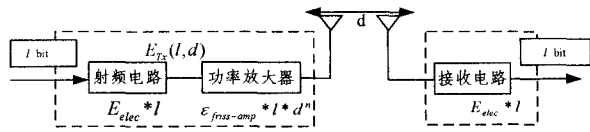


图 4 能耗模型

功率放大器的能耗根据发送节点和接收节点之间距离的不同可以采用自由空间模型和多路径衰减模型。自由空间模型中,路径损耗指数为 2,即 d^2 功率损耗;多路径衰减模型中,路径损耗指数为 4,即 d^4 功率损耗。功率控制通过适当设置功率放大器来转化这种损耗,即如果距离小于门限值 d_0 ,则使用自由空间模型;否则,使用多路径衰减模型。因此,当发射 l bit 数据且传输距离为 d 时,发送方能耗为

$$E_{Tx}(l, d) = E_{Tx-elec}(l) + E_{Tx-amp}(l, d) \\ = \begin{cases} lE_{elec} + l\epsilon_{friss-amp}d^2, & d < d_0 \\ lE_{elec} + l\epsilon_{murray-amp}d^4, & d \geq d_0 \end{cases}$$

而接收 l bit 的消息,接收方能耗为

$$E_{Rx}(l) = E_{Rx-elec}(l) = lE_{elec}$$

E_{elec} 表示每比特数据在发射电路或接收电路中的能耗。

2.3 网络接口

Wireless-phy.cc 定义了网络接口层,它主要完成物理层功能。当此层接收 MAC 层的数据时,根据与接收方的近似距离设置发送功率,减去发送数据所需能量然后将数据发送到信道。如果节点发送完数据后能量被耗尽,则节点死亡且该节点将不接收或发送任何数据并从信道移除。接收数据时,数据从信道进入网络接口层。如果节点处于睡眠状态,则网络接口层丢弃数据;否则,网络接口层设置数据的接收功率。这里会设置一个门限发现功率 $P_{r-detect}$,如果接收功率低于此值,则数据将被丢弃,节点也无法感知数据被发送。如果接收功率大于发现功率且小于接收门限功率 $P_{r-thresh}$,则数据被标识为错误信息并压入站内。如果高于接收门限,则成功接收数据并传送到 MAC 层。

2.4 MAC

数据的发送和接收在 MAC^[10,11] 层是紧密联系的,有必要将 MAC 层对象设置为双向的。以发送方来说,MAC 对象负责给数据包加上 MAC 头部,并将 MAC 帧送给 channel。以接收方来说,MAC 对象从物理层分类器接收位流,并执行 MAC 协议规定的操作后交链路层对象 ll。为完成传感器网络的 MAC 支持,NS 扩展了新的 MAC 协议,定义在 mac-sensor.cc 和 mac-sensor-timers.cc 中。mac-sensor 结合了载波监听多路访问(CSMA)、时分多路复用(TDMA)和一种简单的直接序列扩频(DSSS)模型。其中,TDMA 在 application 层向 agent 层发送数据时在 application 层完成,这个过程称为一个具体的 TDMA 时隙;CSMA 在 C++ 类 macsensor 执行;DSSS 在 application 和类 macsensor 执行。

3 NS2 的无线传感器网络路由协议

3.1 协议结构

图 5 是经典分簇路由协议 LEACH^[12,13] 在 NS2 下的执行结构图,协议运行的环境配置用 Tcl 脚本语言实现。程序开始运行的是入口文件 wireless.tcl,该文件接着引入了移动节点类文件 ns-mobilenode.tcl 和 leach 路由协议文件 leach.tcl。在 leach.tcl 设置了一些仿真参数并引入了 leach 应用协议类文件 ns-leach.tcl 和 uamps 工程文件 uamps.tcl。在 uamps.tcl 文件中又引入了资源敏感节点类文件 ns-ranode.tcl、基站管理类文件 ns-bsapp.tcl、辅助函数文件 extras.tcl(节点间距离的计算)、统计信息函数 stats.tcl(如节点能耗、基站接收到的数据量以及节点存活情况)、资源管理类文件 ns-resource-manager.tcl、能量资源类文件 ns-energy-resource.tcl、邻居资源类文件 ns-neighbor-resource.tcl。

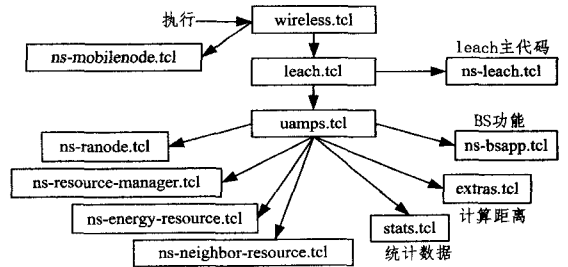


图 5 协议执行结构

3.2 评估标准

虽然无法在现实环境中布置传感器节点网络并进行性能的测试分析,但运用仿真工具后也可以选取部分评估参数对协议的性能做定性分析。从上面的协议结构中可以看到,脚本文件 stats.tcl 是一个专门作为统计数据而存在的文件,它统计了节点的各项参数。协议的性能评估参数选取以下5种。

1)成簇个数:合理的成簇数目可以保证路由协议在网络中运转更长时间。

2)簇头分布:簇头分布密集程度的合理性可以降低个别节点的能耗而增强网络连通性的概率。

3)簇头能耗:簇头在协议中要充当数据融合和路由的作用,簇头轮换机制可缓解簇头能耗过快。

4)建簇能耗:协议的大量能耗发生在数据传输阶段,此阶段能耗相对较小。

5)节点存活量:网络的部署是以节点存活为基础的,网络要能正常运转,节点能耗必须控制。

3.3 仿真实例

仿真环境配置如下:网络区域为 $200\text{m} \times 200\text{m}$,区域左下角为坐标原点(0,0),传感器节点数为200个且随机均匀分布在网络区域中,基站布置于(100,350)处且距离门限值 d_0 设置为87m,簇头轮换时间为20s,自由空间衰减参数为 $10\text{pJ}(\text{bit} \cdot \text{m}^2)$,多路径衰减参数为 $0.0013\text{pJ}/(\text{bit} \cdot \text{m}^4)$,节点的初始能量为0.5J,数据包大小为500bytes,收发电路的功率损耗为 $50\text{nJ}/\text{bit}$ 。以3.2节中的评估标准,在NS2下对LEACH这种典型分簇结构的路由协议进行了仿真。

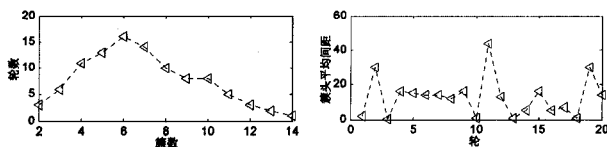


图6 LEACH的成簇个数分布

图7 簇头间的最小距离

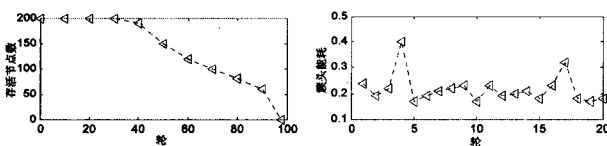


图8 节点存活情况

图9 簇头能耗

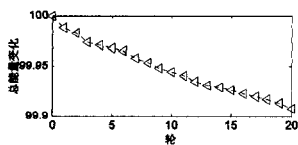


图10 建簇能耗

仿真过程中,NS2下生成的以下几个文件在不断更新以记录相关数据:leach_alive,记录不同时间节点的存活情况(0表示节点死亡,1表示节点仍存活);leach_energy,记录不同时间节点的能耗情况;leach_data,记录不同时间节点发送的数据情况;leach_out,记录整个仿真的完整过程,包括导入的节点拓扑结构信息(节点位置),仿真开始后各节点是否当选

簇头信息,当前轮各节点的簇头信息,标识,距离,节点发送的数据和簇头接收的数据,每轮结束时网络总能量、总数据量、总存活节点数等信息;TDMA_schedule*.txt,记录每轮的成簇及成员调度情况,*表示仿真进行的时间。通过对以上文件的分析,可以得出LEACH协议的各项性能,如图6—图10所示。由于LEACH协议本身成簇算法的限制,图6、图7和图9反映出LEACH成簇时缺乏稳定性,成簇数目无法根据当前节点部署状况形成较优化的簇分布。图8说明LEACH在40轮以后节点死亡较快,随着协议运行,能量受限的传感节点的工作也得不到保障。图9说明LEACH协议成簇时消耗能量较少。

结束语 本文分析了NS2的仿真机制,重点分析了分裂对象模型和离散事件调度模型,然后从节点能耗模型扩展、移动节点扩展以及相应路由协议扩展详细剖析了传感器网络各层的扩展机制,并以分簇路由协议LEACH为例,分析了传感器路由协议的构件代码库的添加机制,并进行了仿真实验,为NS2模式下新协议的代码扩展提供了参考思路和解决方法。

参考文献

- [1] 方路平. NS-2网络模拟基础与应用[M]. 北京:国防工业出版社,2008
- [2] 张铭,宴赫蕾,常春藤. OPNET Modeler与网络仿真[M]. 北京:人民邮电出版社,2007
- [3] 厂为人,黄河,鲜晓东,等. OMNET++与NS2在无线传感器网络仿真中的比较研究[J]. 计算机科学,2008,35(10):53-57
- [4] Global mobile information systems simulation library [EB/OL]. <http://pcl.cs.ucla.edu/projects/glomosisim/>
- [5] 颜昕,李腊元. NS的仿真机制及协议扩展[J]. 武汉理工大学学报:交通科学与工程版,2004,28(2):182-186
- [6] 石硕,顾学迈,张文彬,等. 移动Ad-hoc网络的NS2仿真机制及代码分析[J]. 计算机工程与设计,2008,29(18):4639-4643
- [7] Heinzelman W. Application-Specific Protocol Architectures for Wireless Networks[D]. Massachusetts Inst. of Technology, June 2000
- [8] Heinzelman W, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocols for wireless microsensor networks[C]//Proceedings of the Hawaii International Conference on Systems Sciences. Jan. 2000
- [9] Heinzelman W R, Chandrakasan A, Balakrishnan H. An Application-Specific Protocol for Wireless Microsensor Networks[J]. IEEE Transactions on Wireless Communication, 2002, 1(4): 660-670
- [10] 郑国强,李建东,周志立. 无线传感器网络MAC协议研究进展[J]. 自动化学报,2008,34(3):305-315
- [11] 李春时,王光兴. 无线自组织网络中的IEEE802.11MAC协议的研究[J]. 计算机科学,2007,34(11):26-28
- [12] 张伟华,李腊元,张留敏,等. 无线传感器网络LEACH协议能耗均衡改进[J]. 传感技术学报,2008,21(11):1918-1922
- [13] 汤强,唐晓颖,汪秉文. 簇头预测分布式层次路由协议[J]. 计算机科学,2010,37(2):78-81