TELKOMNIKA, Vol.11, No.1, January 2013, pp. 103~114
ISSN: 2302-4046

Hardware-in-the-loop and Parallel Simulation Architecture for Wireless Sensor Network

Shihong Duan*, Yadong Wan, Peng Meng, Qin Wang
School of Computer & Communication Engineering
No 30 Xueyuan Road, Haidian District, Beijing, 13701323176/
*corresponding author, e-mail: duansh@ustb.edu.cn

Abstract

Discrete event-based simulation is commonly used to evaluate research on Wireless Sensor Networks (WSNs). However, highly accurate simulation models are required in recent advances on wireless communication technology, which results in a steep increase in simulation complexity and runtime. The contributions of this paper for the are twofold, one is to present a general layer structure for hardware-in-the-loop emulation and WSN simulation embedded with implementation of models, such as energy model and link mode, to introduce the distributed nodes into the simulation framework; the other is to build a parallelized simulation based on multi-processor computers as the de-facto default hardware platform and powerful private computing clusters to mirror the real WSN more closely. The work in this paper is realized and used to simulate industrial WSNs for describing and verifying the detail and methodology of WSNs.

Keywords*: Event-based simulation, hardware-in-the-loop emulation, parallelization, energy model, link model*

1. Introduction

The models in simulation of wireless sensor networks become more complex to accurately describe wireless channel characteristics, nodes' operation and the properties of transmission technologies, which typically lead to a high computational load and extensive simulation runtimes. Also, the practical wireless sensor network of large scalbility includes more than 1000 nodes, as objects in simulation, to consume considerable memory and CPU time. So reducing runtimes with building intensive node model and enviorment model in simulation is the key issue of this paper.

Currently, wireless sensor network simulation tools are divided into protocol simulator and program code simulator. The former is represented by OMNet++, OPNET and NS-2, which normally defines the protocol stack in a node and emulates the data transmission in the network. The latter is represented by TOSSIM, which describes operations of single node in high-precision to accurately imitate application behaviors of nodes. The features of frequently used simulation tools are listed in table 1. Comparatively speaking, OMNet++, which is open-source suits for larger-scale network simulation, supports simulating network protocol and debugging code. So simulation tools in this paper are developed based on OMNet++ and extended with models and layed architecture of node and network.

Table 1. The Comparation of Simulation Tools

Tools	Accuracy of network	Accuracy of node	Simulation scale	Channel model	Energy model	Object-oriented	Application
OMNet++[1]	High	Low	Larger	Simple	Simple	Yes	Protocol and debugging
OPNet[2]	High	Low	Larger	Simple	Simple	Yes	Protocol
NS2[3]	High	Low	Large	Simple	Functional	Yes	Protocol
SENSE[4]	High	Low	Large	Non	Functional	Yes	Protocol
TOSSIM[5]	Low	High	Large	Non	Functional	No	Debugging

Received September 28, 2012; Revised November 22, 2012; Accepted November 29, 2012

We make the following four contributions in this paper:

1) We propose a set of generic simulation architecture to define nodes and WSN. Layed structure is realized with independent components cooperated with each other by messaging and sharing data in bulletin board.
2) Link reliability model from industrial enviorment and energy model are integreated into simulation framework to verify the efficiency of WSN technologies more crediblely.
3) A mapping scheme is applied to build up hardware-in-the-loop emulation, in which real nodes send operating parameters to simulation platform via data collection network and work as virtual components in simulation.
4) A parallel architecture deployed in multi-core computer with centralized scheduling is proposed to simulate larger WSNs with layer topology and more than 1000 nodes.

Also, we use the said platform to simulate a typical industrial application compliant with WIA-PA standard. Our evaluation shows the correctness and availability of simulation platform, and parallel simulation and hardware-in-the-loop emualtion improve performance with less running time and larger scale WSNs over traditional scheme.

The remainder of this paper is structured as follows. Section 2 gives a introduction to our simulation frame work as solution of less capacity and more running time of traditional simulation techniques and detailed design and realization of architecture, models, hardware-in-the-loop emualtion system and parallel simulation system. Section 3 analyzes synchronization scheme in terms of the granularity of the parallelism and simulation time, verify the model accuracy and evaluates the effectiveness by an simulation instance of industrial WSN application. Finally, we conclude in Section 4.

2. Research Method

This section introduces the fundamentals of our simulation framework PIWSNSim and its underlying model realization, function component mapping tactics and parallelization scheme. PIWSNSim enables a parallel execution of network simulation models to reduce running time by means of three properties: 1) It introduces a layed structure to easily describe node, network and enviorment. 2) It defines a mapping scheme from real node to vitual component in simulation to realize a hardware-in-the-loop emualtion, even code debugging. 3) It employs a parallelization scheme to execute independent simulation events simultaneously, control by centralized scheduler. PIWSNSim enables a credible simulation by building up more accurate link realibility models and energy models of node and network.

2.1. Time-varying Link Reliability Model in PIWSNSim

Many industrial WSNs deployment is underway nowadays to allow the engineers to reliably acquire and control the real-time data of WSNs in the factory at any time anywhere. The noise and interferences are significant due to the wide operating temperatures, strong vibrations, airborne contaminants, excessive electromagnetic noise caused by large motors etc. Therefore, it is important for a simulation framework to playback characteristics of communication channel in order to verify the technology and optimize the design of industrial WSNs. IEEE 802.15.4 based WSNs are designed to support communication over short ranges with low data rate and reduced energy consumption. So the IEEE 802.15.4 standard has become a recognized industry standard and well accepted by industrial users. [6] measured channel characteristics of sixteen IEEE802.15.4 radio channels in typical industrial environment. It is found that channel reliability is possibly and dramatically different with each other and none of the channels can always provide good reliability required by industry application. Furthermore, reliability of each channel varies with time and space.

Definition 1. PDS, Packet Drop Sequence, is the log of the success or failure in one time-slot communication based on ACK received or not, shown as Eq. 1, wherein i denotes the ith transmission. PDI, Packet drop interval, is the time gap between two adjacent lost packets to define the PDS distribution.

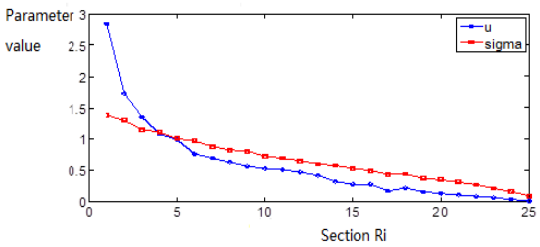
$$PDS(i) = \begin{cases} 0 & received \quad ACK \\ 1 & others \end{cases} \tag{1}$$

Definition 2. PDR, Packet Drop Ratio is the indicator of lost packets numbers without ACK identified by the sender in the log window of T including N transmission numbered from j to j+N-1, shown as Eq. 2, wherein Numberpacket is the total sending number of WSN in T period, and NumACK is received ACK numbers.

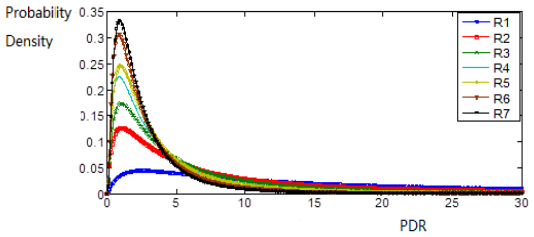
$$PDR = \frac{NumPacket - NumACK}{NumPacket} \times 100\% \quad (2)$$

From test-bed measurement, PDI follows same distribution of lognormal in some periods of (ti,tj] with different property parameters based on PDR value. PDR is divided into 25 section R1~R25 with value [1-4%], [5-8%], … , [97-100%], and probability density distribution of PDI is defined in Eq. 3, where u and σ have different value in different PDR sections shown in Figure 1(a). Also the PDI distribution curve is shown as Figure 1(b).

$$f(PDI) = \frac{e^{\frac{PDI-\mu}{\sigma}}}{\sigma\left(1+\frac{PDI-\mu}{\sigma}\right)^2} \quad (3)$$



(a) u and σ value from measurement



(b) PDI probability density distribution in R1-R7

Figure 1. Time-varying model in industrial application (expressed in PDR, PDI, and PDS)

PIWSNSim as a simulation framework for WSNs builds the said time-varying link model shown with an independent structure named as EIF shown as Figure 2. EIF can be not only largely implemented in “pure” OMNeT++; but also successfully incorporated into the MiXiM[7].

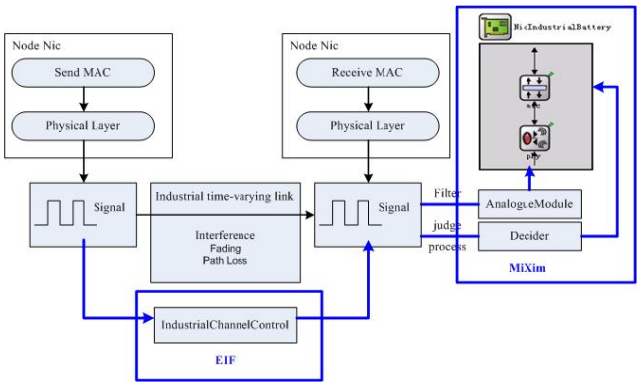


Figure 2. Time-varying ink model of EIF in PIWSNSim

MiXim framework provides channel model to filter and process signal in physical layer transmission. The AnalogueModels are responsible for simulating the attenuation of a received

signal, such as shadowing, fading and path loss. The Deciders are responsible for classifying the signal as noise or signal, and calculating bit error of the received messages. Reliability of channel in 802.15.4 based industrial WSNs varies with time due to all kinds of factors, such as multi-path interference, wireless radio interference and so on. Time-varying characteristic of channel can be expressed as oscillations in short period and mutation when some event happens in industrial fields. By fit test using dfittool in Matlab, oscillation in small window and interval between two events all follow the log-logistic distribution with different parameters; Probability density distribution of affected channel numbers and affected continuous frequency channel number are all keep to the log normal distribution with different parameter values. EIF is responsible for simulating the signal loss in typical metallurgy process industrial applications.

EIF provides a separate simple module named as IndustrialChannelControl to describe the loss rate of 16 frequency channels. IndustrialChannelControl possesses the scalable and extensible mechanism.

1) The operation parameters can be configured. The parameters are provided in the xml file, and the xml file name is specified in the simulation configuration. So users are easy to adjust parameters according to channel characteristic in diverse practical applications.

2) The model can be replaceable. Currently, test from three typical industry fields introduces the idustrialChannelControl which will be expanded, improved or substituted for more industry environment. IndustrialChannelControl in essence is a c++class; other models can derive from it. Industrial link model name is handed over to the physical layer, so it is easy for physical layer can make decision which link model it calls.

2.2. Layed Architecture of PIWSNSim

PIWSNSim is designed to describe elements in WSN, including nodes, enviorment, and management units. Layed simulation architecture is employed to simulate WSN in detail and reality by clearly defining elements in WSN with modular mode. Shown as Figure 3, PIWSNSim based on OMMet++ platform emphasizes the importance of environmental modeling and node modeling. Four layer structure combines nodes in resource layer with wireless medium defined in media layer under the regulatory units in manager layer to simulate the WSN application meticulously, and the network operation and performance analysis are exhibited to user by friendly GUI layer. PIWSNSim uses discrete event handling mechanism provided by OMNet++, manages event queue pushed by components of WSN and distributes events to CPU kernel.

PIWSNSim in this paper is realized as a composite model defined in OMNet++ and called by Sim kernel. The components in layed structure are indepent and can be easily replaced and updated to help researchers establish WSN simulation for specific applications, and analyze technology and algorithm for their concern. The layed architecture employed in PIWSNSim are narrated as follows.

1) Resource layer includes all nodes in the network. Nodes are defined as hierarchical structure to support description of isomorphism and heterogeneous nodes.
2) Media layer includes modeling wireless transmission channel and sampling characteristics. Link reliability model described in section 2.1 is realized as wireless link model in media layer to accurately analyze WSN technology.Sensing model with sampling enviorment model are used to simulate sensing process and characteristics of system on test.
3) Management layer takes responsibility of setting the network properties and building WSN to support self-formation and self-maintenance of WSN, including management units of topology, synchronization, connection, security scheme, information logger to make it easier to do research on related technologies of improving WSN performance.
4) GUI layer provides users with friendly graphical interface and replace or extend windows interface provided by OMNet++. Network configuration, online or offline analysis and forecasting of simulation data are realized in GUI layer. GUI application can be not only called by sim kernel, but also can run alone to read/write configuration file and select data from database to give performance enhancement report.

Nodes in PIWSNSim platform should have same function with real node to sample, compute and communication. So simulation nodes are supposed to have a clear framework to support change or control of the network topology and desgin algorithm. A stack structure for a node is proposed in this paper shown in Figure 4(a). Node with stack structure consists of a number of different function stacks that are derived from the same template stack. Functions of communicating, sampling, energy managing and positioning are respectively implemented in

specific stack. The highest level of each stack is a shared application layer. Stack structure in this paper brings the benefits of the network protocol stack into describing related key tasks, such as code reuse, standardization enhancment and interchangeable protocol design.

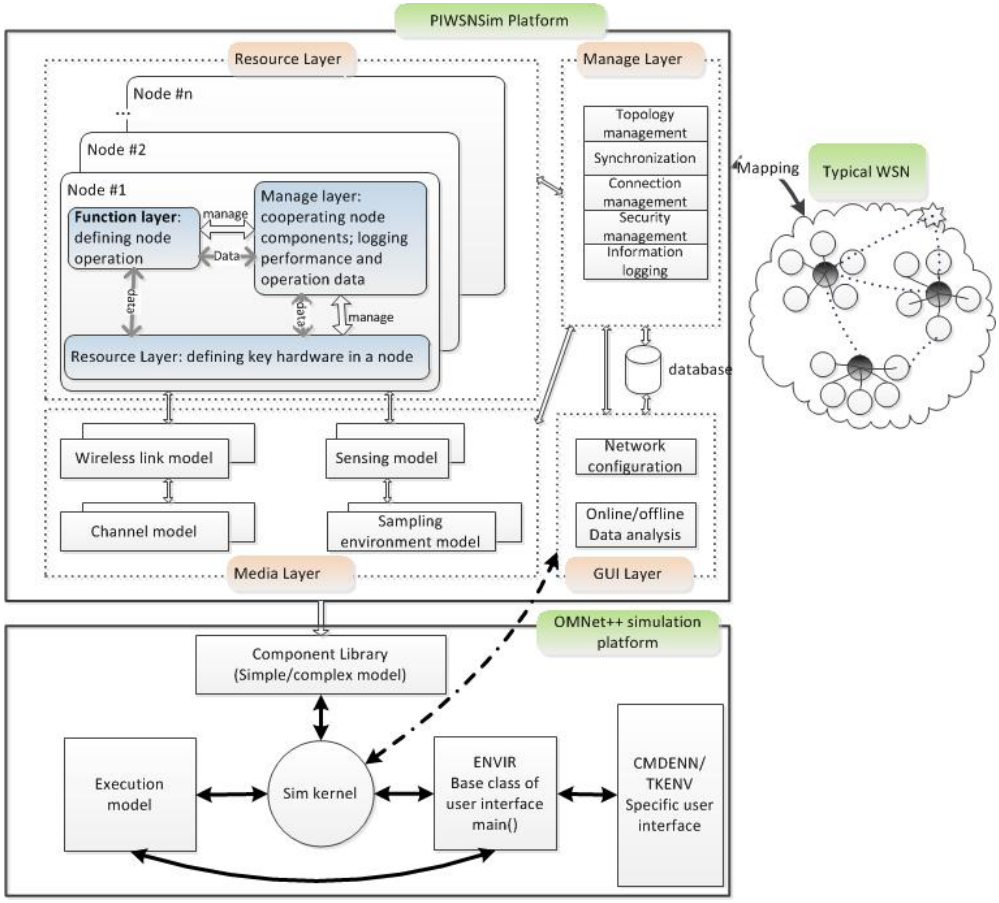
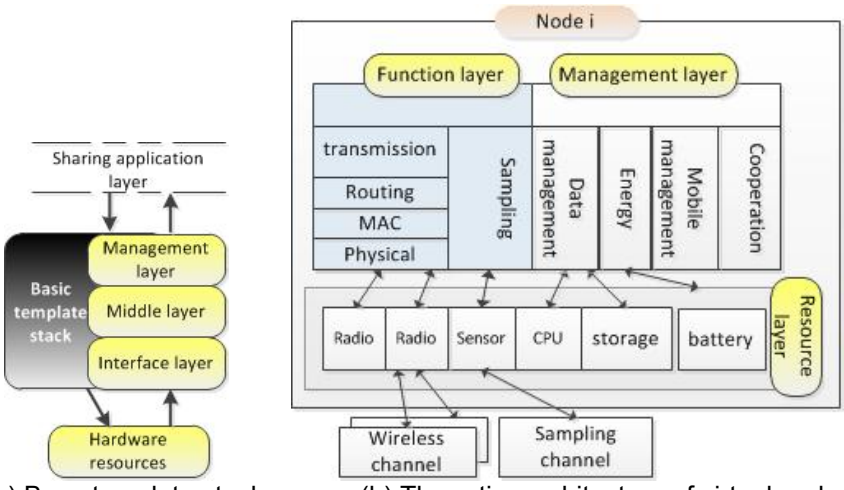


Figure 3. The Layed architecture of PIWSNSim



(a) Base template stack (b) Three-tier-architecture of virtual node

Figure 4. Virtual node defined in PIWSNSim

The virtual node in PIWSNSim is a composite component class written in c++ code to simulate work process of real sensor nodes with the embedded software. Also the virtual node is a complex module in OMNet++ environment. In order to facilitate the implementation of different simulation senario and switch between different functions, the virtual node adopts a three-tier-architecture architecture partitioned by hardware resouces and software function, shown in Figure 4(b), including layer of function, management and resource.

Resource layer is responsible for establishing models of node hardware resources and describing the state change of hardware resources and energy consumption. The common hardware components include a power supply module which is normally a battery module, a storage unit, a CPU, several sensing interfaces and several RF modules. The battery module provides discharging model of some battery and records the change of the working state of the battery by powering to other resource members. RF module implements signal transceiver function. Transmission signal attenuates in accordance with the time-varying model defined by radio channel and be sent to the RF module of target node via a wireless channel. Sensing interface send sample messages to sensing physical channel. It simulates the collection of physical information based on the model defined by sensing physical channel and generates the characteristical sample data.

Management layer takes responsibility to coordinate the work of the various components in a virtual node for the specified application tasks. Management modules realized in PIWSNSim includes energy management, mobile management, data management and collaboration management. Collaboration management provides a bulletin board to which various components in node could register message type responded by theirsleves. Specific message will be issued by source component to bulltin board, and then be transfered to the components that subscribe the message, finally the components execute the response operation. Bulletin board module provides a unified interface for different tasks to use resources. Energy management unit monitor and analyze energy to support the simulation of energy-aware strategy. Mobility management defines a large number of mobile model, virutal node moving according to specified mobility model regularly and notify connection management module to re-establish the network connectivity and update the topology based on the coverage attribute of the node radio frequency module. Data management establishes data storage and fusion model, providing management strategies for related applications.

Node function is responsible to define node operation and simulate workflow of physical nodes. The task of vitual node consists of transmission, sense and computation. When transmitting data, the application layer will send the data to routing layer, MAC layer and physical layer, and finally notify the RF module to complete the signal transmission. Signal from radio channel will be received by the RF module of the nodes in the coverage and upstreamed to the application layer sequentially. When sampling data, application layer will notify sensor driver, then to sensing interface to acquire physical information. At last, sampling data are collected and issued to the application layer. Because computation task is to do various operations of data, so computation model is described as CPU state transition diagram used in data management and energy model.

The base class of template stack is stratified multi function layer. Each layer, independent from other layers, contains enough manipulation tasks and coordinates through the appropriate interface. Levels of the stack are defined as follows:

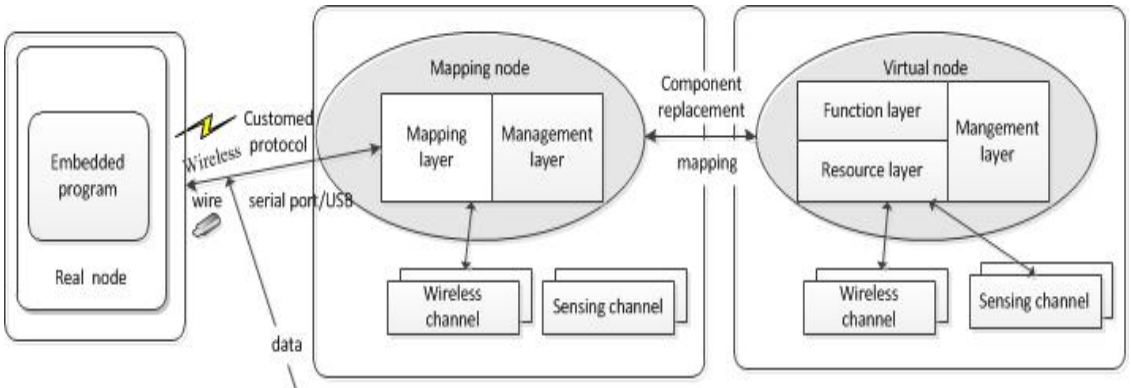
1) Interface Layer is responsible for communication with the appropriate hardware and make complexity of implementing the hardware functions invisible to upper module. For example, in communicaton tasks of the node, interface layer indeed is physical layer.
2) The middle layer provides low-level function of error checking, queue scheduling and etc to wok as relay of data flow and control flow. The MAC layer in communication task is a typical middle layer.
3) Management layer provides high-level function of event detection, network formation and self-maintainence to work as the task controler. Routing layer in communication task is a typical management layer.
4) Shared application layer coordinates the collaboration between different stacks.

The three lower layers of the stack structure respectively works as logical interface, processing unit and controlling unit. Architecture with shared application layer takes into account the characteristics of the modern WSN communication protocol and general parallellizaiton.

2.3. Design of Mapping Scheme Between Real Node With Virtual component

In order to reduce running time of a large scale WSN simulation and also debug the embedded code in sensor nodes, hardware-in-the-loop emulation platform is extended from the layer architecture proposed in section 2.2 by using a mapping scheme between real node with virtual component in PIWSNSim. The mapping scheme integrates real nodes into simulation platform as embedded components of virtual nodes. Running parameters of real nodes are transmitted to PIWSNSim; relative enviorment models defined in PIWSNSim provide physical application enviorment simlation, transfer data from some real node to other nodes, and emulate practical WSN. Running information from real nodes to PIWSNSim supports debugging code easily. Real nodes are used in hardware-in-the-loop emualtion to credibly verify the effectiveness of some optimization technology, provide reliable data basis for the actual deployment of the network, and form distributed simulation to reduce running time.

Shown in Figure 5, mapping nodes are deployed in resouce layer of PIWSNSim with two tier are responsible for tracking the actual operation of the sensor nodes and passing running information to wireless link model under the some management strategy to form clustering wireless sensor network which transmits data pakcage from some sender node to other receiver nodes. Sensor nodes send running state to mapping node via wireless method, such as using network protocol stack to communicate with simulation virtual node, or wired method, such as the serial interface or USB interface. Mapping node replace the some function of virtual node and build up the mapping between real nodes and virtual nodes. Multi communications modes used between physical nodes and virtual nodes in PIWSNSim and multi-core and multi-thread mechanism provided by PIWSNSim platform supports large-scale network simulation with more than 1000 nodes.



Data type	Data format				
Network related data	Node ID	Static data: location	Dynamic data: surplus energy	sampling data	Target node: broadcast/unicast
Running state data	Node ID	Node function ID	Debugging data: value of key varibles		

Figure 5. Mapping scheme and data defition in PIWSNSim

Real nodes send network related data to mapping node with two tier and work as a component in PIWSNSim by such mapping scheme to replace task layer and resource layer defined in simulation nodes. The management layer are kept in virtual node to simulate some monitoring or cooperation strategy. Data stream between real nodes and virtual nodes are divided into two classes. One is network related data send among real nodes, and the other is running state data for code debugging. The network related data is a quintuple of (node ID,)

The interfaces of different layers, nodes and models are defined unified. The components in different layers can be easily assembled and dismantled in the system. Simulation steps to implement hard-in-the-loop emulation are descibed as follows:

1) Buiding a new component of mapping layer in mapping node to replace the function layer and resource layer defined in virtual node shown as Figure 5. The interfaces between mapping layer and management layer or wireless channel are unified to use the simulation models easily.
2) Mapping layer in mapping node take responsibility for connecting real nodes via wireless or wire mode, and catching the real node running status, tracking/debugging nodes embedded functions to map real node to the virtual node. Also simulation models employed in mapping nodes integrate real node into PIWSNSim virtual network.

2.4. Parallelization Event Scheme on Multicore Computer

Multi-processor computers[8] constitute the de-facto default hardware platform even for desktop systems to provide cheap yet powerful "private computing clusters". So the parallelization of discrete event simulations significantly gained importance and can be put into earth to reduce running time of large-scale WSN simulation. This paper presents a simple dynamic centralized parallel scheduling method to support the simulation of large-scale WSN by scheduling independent event produced in simulation progress. Queue of time-stamped event with time modules is created and maintained dynamically by related components. A parallel event scheduler proposed in this paper takes reponsibility to manage event queue and distribute event to some core of computer.

Parallel simulation in PIWSNSim includes modeling execution time of each event, and dynamical centralized event scheduling scheme. The steps of parallel simulation is shown as follows:

1) Embedded software of WSN nodes are designed and developed according to application requirements of WSN.
2) Event execution time modeling method is employed to define static minimum time of event, and the dynamic run-time correction. Key value of the time model are recorded in the xml data files.
3) Simulation mode is confirmed. If simulation is needed, virtual node with three tiers is designed; if emulation is needed, mapping node with two tier is built up by mapping scheme.
4) Simulation WSN is established following the application needs, including to define the network topology, the channel model, sensor model and so on.
5) Event scheduler use the the dynamic centralized parallel event scheduling method to simulate the operation of large-scale WSN and analyze performance. Event scheduling method in this paper is parallel execution strategy to initiatively issue event to various core and optimized schedule, which is proposed for computer of multi-core processor with shared memory.

The event scheduler is core unit to achieve centralized management, dynamic optimization. Event in simulation successively be put into pending execution event queue; each CPU core has a worker thread and maintains a storage unit holding event processing tasks. Memory unit of executing event of task is set a spinlock to provide safe sharing service in multi-core processor system. Function components execute task and management, produce time-stamped events and push event to pending event queue, which is protected with mutex semaphore or conditional variables. Figure 6 describes the schedulling process controled by centralized scheduler. The solid line is event stream, and the dashed line is the synchronous data stream.

Key elements in parallelization are defined with sets, variables and functions. E={e| e is event produced in simulation models}, F={e|e is some pending event}, D={e| e is issued event to CPU core by scheduler}, W_i={t | t is worker thread in ith CPU core}, and F⊆E. Each event has two property value {T_{start}, T_{end}}, wherein T_{start} is initial event execution time and T_{end} is event end time. Twin is s synchronous window of current event. Task = NULL means memeory unit of executing event is empty. Function of waitForOneThread is defined to query if there is idle thread, if not the execution is blocked until some thread is idle which is processed as return value. The scheduling algorithm is designed as follows.

1) During initialization, function components push static events into F, and $T_{win}=\infty$。
2) Event scheduler extract the first event in F as current event as. Then T_{win} is updated by comparing T_{start} of $E_{current}$ and T_{win}. If $T_{start} \leq T_{win}$, e can be executed parallel with $E_{current}$, else e will be wait until the value of T_{win} is changed. CheckParallization function judges the simultaneity of event.

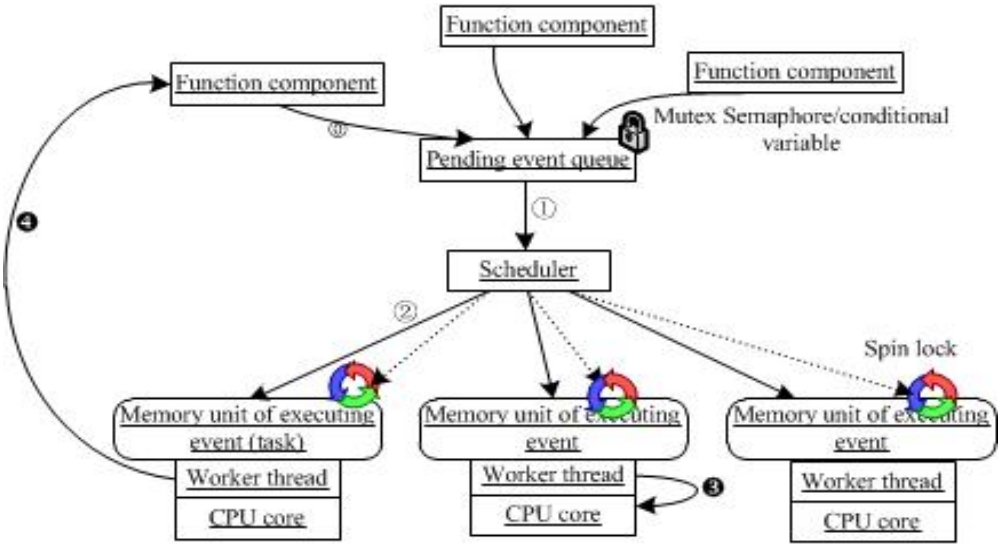


Figure 6. Scheduler processing in PIWSNSim

```
checkParallization(){
  do{
   e=min{T_starte'|e'∈F}
   if (T_starte≤T_win)
        return (e)
  else
        waitForSomeThread(w|_Twin.job=NULL)
  T_win=min{Tend(e')|e'∈D}
  }while(true)
}
```

3) Scheduler queries the status of worker thread to find the available one, moves $E_{current}$ to set of D, distributes $E_{current}$ to availabe worker thread, and finally change the value of $T_{win}=\min\{T_{end}(e)|e\in D\}$。
4) Worker thread checks the local event status, that means the status of memory unit of executing event of task. If task if free, the event will be processed; else task is spinlocked to synchronize threads.
5) Worker thread executes event process program to generate new events marked with time stamp and pushed into F. Once the worker thread is finished, the value of $T_{win}=\min\{T_{end}(e)|e\in F\}$.

Centralized scheduling architecture with management of dynamic events controled by event scheduler can balance loads effectively. In WSN simulations, mobility of nodes, time-varying characteristics of channel can bring unpredicable chagen of work load. Centralized management scheme will be well adapted to sudden alteration during runing without additional balance strategies.Scheduler takes responsibility to query event queue in sequence, and find out the current event, and distribute event after checking the parallization.

3. Results and Analysis

This section presents evaluation of correctness the link reliabiility model, effectivess of parallelization scheme, and at last uses PIWSNSim platform to simulate a practical industrail WSN with 400 nodes to analyze the extensibilty of our simulation and emulation framework.

3.1. Link Reliability Model (EIF) Analysis

Two simulations are designed to test the correctness and validity of EIF. First one is to produce PDS according to model based on offline test samples in our lab by using EIF and OMNet++. Meanwhile a practical test is executed between two nodes which recorded their own

PDS and another simulation base on Two-Ray Ground channel Model with error ratio of 10% is implemented by using NS2. The analysis shown in Figure 7 proves the link reliability mode in PIWSNSim is more correct.

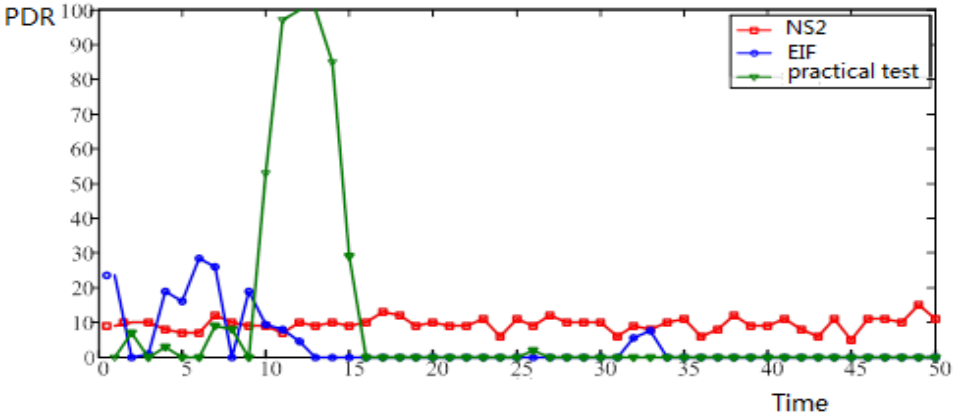


Figure 7. Correctness result of link reliability model compared with NS2 and practical test

Second simulation is to test validity and scalability of EIF by implementing the specific hopping scheme. Channel redundancy is one of the techniques to increase the reliability and determinism of data communication because none of the channels can provide a good reliability required by industry application. Slotted hopping and slow hopping are always used in industrial application. In slotted hopping, channel-hopping timeslots occupy equal durations. Taking account of link characteristics to increase the reliability, some adaptive frequency hopping methods have been proposed to satisfy the high reliability requirements. That is using the stable and reliable channel without hopping until the channel goes to bad, and then choose a new channel with high reliability to reach a high reliable network communication. This kind of hopping is called adaptive frequency hopping[9].

Table 2. Reliability of AFH and time-slot hopping in single cluster start WSN

Nodes	1	2	3	4	5	6	7	8
AFH (%)	98.1	96.9	94.5	98.0	95.4	100	94.5	96
Time-slot hopping(%)	80	81	79	80	80	80	81	80

EIF provides the both hopping strategy to increase the reliability and TDMA MAC protocol to avoid interferences among nodes and verify the transmission delay. Hopping mode is expressed as a configured parameter in EIF for users to choose which hopping method is called by TDMA MAC[10] module in their simulations. The conclusion is that the adaptive frequency hopping maintains the high reliability due to switching to the channel with high reliability communication shown in Table 2.

3.2. Time effectiveness of parallel scheme in PIWSNSim

An instance simulation is implemented to verify the time effectiveness of parallel scheme introduced in Section 2.4. The simulation playground is 1000m*1000, and there one sink node located in coordinates of (0,0), other nodes are randomly placed on some location in the playground. Nodes consults with neighbors in accordance with the location to self-organized into clusters with less than 64 member nodes.

Nodes are equipmented with network protocol compliant with IEEE 802.15.4 standrad to upload data regularly. Cluster heads are selected according to the distance from sink node and node ID. Topology of a cluster is star structure, and TDMA mechansim is employed to send

data from member nodes to cluster head. After heads collect the data of the cluster, then adjust to unified specific frequency and adopt tree routing scheme to transfer data to sink node. Models used in simulation are shown in Table 3.

Table 3. Parallel scheme simulation definition

Definition	Simulation Models
Topology	Cluster network, start and mesh
MAC layer	TDMA, CSMA/CA
Phsical layer	OQPSK demodulation
Radio module	CC2420 RF with status transition
Channel model	free space model
Sampling model	No
Mobility model	Stationary(Mobility= NULL)

PIWSNSim runs on A840r-G server with 12 AMD cores. Nodes number n is selected respectively as n=100, 200, 500, 800, 1000, 1200, 1500,1800,200. Simulation are executed in three modes, no parellim scheme, regarding node as function unit or regarding cluster as function unit to issue independent events. Simulation ends when pratical WSN runs for 1 hour, and simulation execution duration is recorded in value of T_{exec} . The result shown in Figure 8 implies that parallism scheme improve running speed. The duration dramatically reduce to almost 50%.

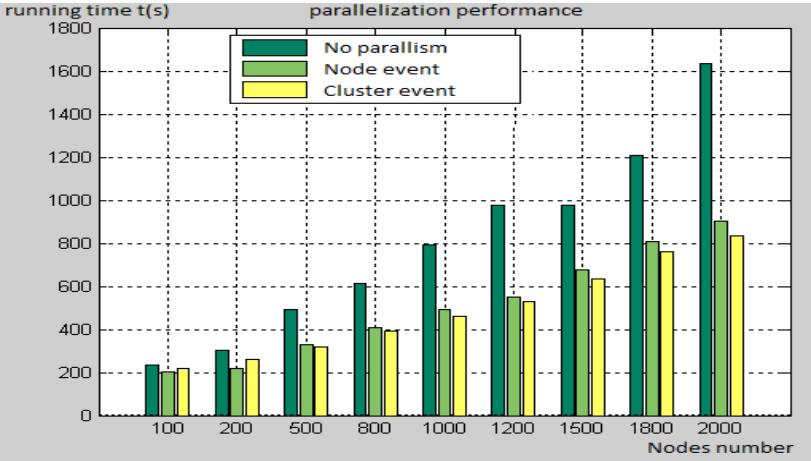


Figure 8. Simulation duration in different paralelization scheme

3.3. Simulation of practical industrial WSN

We use following steps to simulate a practical industrial WSN:

1) Network nodes are defined. Each node is a composite component that will be assembled on basis of the component library provided by PIWSNSim platform, or the gateway and routing node that form a mesh network defined by configuration parameters, or star cluster on site monitoring equipments.
2) Reliability model, a time-varying reliability model under typical industrial environments is defined adn a recursive battery model is defined.
3) Performance indicators are defined to log record various performance-related data and give analysis chart by OMNet++ or GUI utility.
4) Topology management are defined to deploy nodes in the network playgroud and simulate layer WSN.

There are 400 nodes deployed statically on the monitoring unit equipment sampling temperature information in simulation. Cluster heads located in predefine splot are powered by wire supply to build up backbone network of the monitoring network system. So, clusters are

also scheduled with the total number six. The number of nodes in each clusters range from 60 to 80. There is an alarm relay node for alarming in each cluster. The simulation results are analyzed as follows:

1) The actual sampling period of the equipment monitoring system is 20 minutes. The system actually runs for one week. The time of using multi-threaded simulation in single-core machine is about 16 minutes.
2) The energy consumption is balanced in actual operation of the equipment monitoring system. After running for 22 months, the battery voltage drops to 0.6V. After same time simulation, the voltage drops to 0.65V.
3) Reliability reached 99.3% in the actual equipment monitoring system. Simulation gives that simulation average relibility is 99.5, and transmission delay is 5s caused by TDMA time slot allocation scheme and retransmission.
4) Command are issued from GUI utility developed in our lab to forge fake failure, the alarming transmission delay is 0.1s is same with practical test.

4. Conclusion

PIWSNSim provides stack layer architecture of sensor nodes to build up virtual node in simulation, and support mapping from real node to virtual node to simulate or emulate large scale WSN. PIWSNSim extends parallel mechanism provided by OMNet++, including building event queue with the time stamp and resources ID, executing dynamic centralized event scheduling strategy based on multi-core multi-threading technology. Parallism scheme employed in this paper make platform extensible and scalable. CPU time in the simulation verify the effectiveness of parallism schedule based on event management.

Acknowledgement

This work was supported partially by National High-Tech Research and Development Program of China under Grant No.2011AA040101; National Natural Science Foundation of China under Grants No.61003251, No.61172049, No.61173150; Doctoral Fund of Ministry of Education of China under Grant No.20100006110015.

References

[1] Andras Varga, Rudolf Hornig. *An overview of the OMNeT++ simulation environment*. 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops. Marseille. 2008:1-10.
[2] Jie Cai, Bo Jia. *Network Simulation Based on OPNTE and Application.* 2009 First International Workshop on Education Technology and Computer Science. Wuhan. 2009: 199-202.
[3] Teerawat Issariyakul, Ekram Hossain. Introduction to Network Simulator NS2. New York: Springer Publishing Company. 2010.
[4] Rosa PMP, Neves PACS, Vaidya B, Rodrigues JJPC. *G-Sense - A Graphical Interface for SENSE Simulator*. 2009 Advances in System Simulation. Covilha. 2009:88-93.
[5] Mikko Asikainen, Keijo Haataja, Risto Honkanen, Pekka Toivanen. *Designing and Simulating a Sensor Network of a Virtual Intelligent Home Using TOSSIM Simulator*. 2009 Wireless and Mobile Communications. Kuopio. 2009: 58-63.
[6] Wang Qin, Wan Yadong, Li Lei, Duan Shihong. Multi-Channel Reliability Modeling and Analysis for IEEE802.15.4. in Industrial Environment. *Journal of Computer Research and Development*. 2009; 46. (12): 1971-1984.
[7] Nitin G Palan, Aditi P Khadilkar. *Media Access Control protocol modelling for Mobile Sensor Network-using OMNeT++ -MiXiM network simulator*. Sustainable Energy and Intelligent Systems (SEISCON 2011). Maharashtra. 2011:641-644.
[8] Xiaozhong Geng. *Dynamic Load Balancing Scheduling Model Based on Multi-core Processor.* Frontier of Computer Science and Technology (FCST). Changchun. 2010: 398-403.
[9] Wan yadong, Wang qin, Duan shihong, Zhang xiaotong. *RAFH: Reliable Aware Frequency Hopping Method for Industrial Wireless Sensor Networks.* WiCom. Beijing. 2009: 1-4.
[10] T Sameh Gobriel, Daniel Moss´e, Robert Cleric. *TDMA-ASAP: Sensor Network TDMA Scheduling with Adaptive Slot-Stealing and Parallelism*. Distributed Computing Systems (ICDCS 2009). Quebec. 2009: 458-465.