Hindawi
Journal of Electrical and Computer Engineering
Volume 2022, Article ID 8629211, 10 pages
https://doi.org/10.1155/2022/8629211

Research Article

Research and Design of High-Fidelity Experimental Bed System for Wireless Sensor Network

Heqing Huang and Wenjing Liu

Chongqing Technology and Business Institute, Chongqing 400052, China

Correspondence should be addressed to Heqing Huang; huangheqing@cqtbi.edu.cn

Received 22 February 2022; Revised 14 March 2022; Accepted 4 April 2022; Published 18 April 2022

Academic Editor: Wei Liu

Through in-depth analysis of the typical international wireless sensor network experimental system, the current design generally lacks the real-time diagnosis of the health state of the fault node and the experimental platform, the management and control mechanism to minimize the interference of the experimental task, and the highly integrated and friendly experimental entrance, which will have an obvious distortion impact on the experimental results. In view of the above shortcomings, on the basis of fully absorbing the design advantages in this research field, this paper studies and designs a high fidelity WSN experimental system, which covers the elements of open source, heterogeneity, analysis visualization, and experimental repeatability and innovates the mechanisms of real-time diagnosis, hierarchical parallel control, and TDMA to collect a large amount of experimental data. In order to support a variety of applications based on IEEE802.15.4 band, a high-fidelity and user-friendly experimental platform software and hardware system with 15 clusters and 130 heterogeneous nodes is studied and designed, which can simulate outdoor mesh networks with more than 6 hops. This paper is designed and implemented on Linux open-source platform. The whole system has low cost and can be replicated. The design results have good reference and application value for the research and design of wireless sensor network experimental system.

1. Introduction

Wireless sensor network testbed refers to a carefully planned wireless sensor node network designed and deployed in a controlled environment to provide experimental verification under real environmental conditions for relevant technical research [1], as well as parameter configuration before the experiment, various controls, and data collection in the process of the experiment. The technical means such as data analysis and visualization after the experiment are the bridge and important infrastructure for the coordinated development of WSN industry and scientific research.

This paper expounds the basis of this research from two aspects: Firstly, it explains the value of this research from the perspective of the relationship among wireless sensor network industry, technical research, and WSN experimental bed facilities. Secondly, it expounds the research status at home and abroad from the perspective of the research deployment of representative WSN experimental bed in the industry.

The progress of sensing technology has pushed the networking industry into a substantive development stage [2], and the industrial scale will reach several trillion yuan in the next five years. With the acceleration of technological progress and industrial development, wireless sensor network (WSN) has been widely deployed in military, industry, environment, health, and other fields. The resource constraints of sensor nodes determine that efficient protocols and algorithms are the core elements of WSN research, which has aroused the industry's keen attention to WSN high-fidelity experimental facilities.

Wireless sensor network testbed is an experimental platform for WSN research and development. It provides a strict, flexible, transparent, and repeatable experimental test means for scientific research and innovation [3]. Compared with traditional software simulation, WSN testbed has higher authenticity and reliability. The increasingly close cooperation between scientific research and industry also puts forward an urgent demand for the design of WSN experimental bed. Therefore, many international universities and scientific research institutions have invested in the development and deployment of WSN experimental bed to provide an experimental platform as close to the real scene as possible for wireless sensor network research.

As the basic experimental facility of wireless sensor network scientific research in the real environment, the prominent feature of WSN experimental bed is that the environment is real, the experiments can be configured, operated, and monitored remotely, and the repeatability experiments can be carried out by using the configuration parameters to compare and analyze the similarity of the experimental results, which provides a more efficient and fine verification and evaluation means for various algorithms, protocols, and applications [4]. The traditional virtual simulation is purely based on mathematical model, which is more suitable for testing and evaluating the potential characteristics of the new protocol in the theoretical stage, but it is very complex and difficult to abstract its simulation model properly. Because the accuracy of virtual simulation depends on the complexity of mathematical model, it often needs to strike a balance between accuracy and complexity [5]. The higher the complexity is, the more computing resources and time are required, which leads to the tendency of virtual simulation tools to simplify their model design as much as possible and focus on the simulation test of upper layer protocol. Ignoring the collaborative verification between protocol layers in the real environment, it is impossible to bring a variety of channel characteristics in the real environment into the model design. These inherent defects of virtual simulation drive the design and implementation of WSN experimental bed to become an urgent need for the development of wireless sensor network industry and technology research.

2. Related Works

The design and implementation of WSN experimental bed in China lags far behind foreign universities and scientific research institutions, which does not match the current development trend of wireless sensor network industry and technology. In comparison, the research, design, implementation, and deployment of WSN experimental bed abroad have received extensive attention and application. The following representative cases are comprehensively described in three aspects: experimental characteristics, hardware characteristics, and mobile support.

2.1. Experimental Characteristics. In the existing WSN experimental beds abroad, FIT, IOT Lab, TWIST, and WISEBED adopt the interactive mode of web interface for experimental configuration to select the required resources, set parameters, node programming, and data collection [6]. WISEBED and IOT Lab adopt the first-come first-served mechanism for experimental task scheduling. In order to meet the intervention requirements in the experimental process, different experimental beds use SSH protocol interface, VPN, and command-line script to interact with and control the experiment, while Motel uses the precompiled XML file to load the process command. In terms of experimental data storage, files or databases are basically used for storage, and graphical display and analysis tools are provided.

2.2. Hardware Features. In terms of heterogeneous support, TWIST integrates TmoteSky and eyesIFX nodes, while Kansei supports 802.11/802.15.4 and 900 MHz bands and includes three node types: XSM, TelosB, and Imote2. However, most of the operating systems used in the experimental bed are concentrated in FreeRTOS, TinyOS [7], riot, and Contiki. In terms of scale, most of the experimental bed nodes are about 200 nodes, and the large ones can reach more than 700 nodes. However, the experimental bed fusion mode deployed in different regions can be used for scale expansion [8]. For example, SmartSantander has carried out multiregional experimental bed fusion, which is managed by the main server and multiclient mode.

2.3. Mobile Support. So far, most of the experimental beds do not support mobility, and only the way of mobile carrier attaching nodes is used as a mobility supplement to the fixed experimental bed. SmartSantander uses fixed line buses to carry some sensor nodes and gives the moving route and track of nodes for experimental planning in combination with GPS technology, while IOT Lab attaches the nodes to electric toy cars and 200 mobile robots. The experimenter can set the moving speed and mode. Mobility support will also bring difficulties in accurate trajectory positioning and potential collision risk. In addition, the well-designed automatic charging mechanism of mobile carrier is also a challenging problem.

Through in-depth analysis and comparison of the current typical WSN experimental beds, it is found that most of the existing experimental beds generally lack robustness and prediction mechanism design in the case of node failure, and the experimental process lacks user-friendly and highly integrated control intervention support. In terms of resource sharing, the widely used experimental competition mechanism is difficult to meet the needs of timely implementation of experimental tasks. In addition, there is also a lack of an efficient real-time diagnosis design for the key states of nodes (occupation, failure, idle, etc.) to improve robustness. For example, Kansei's experimental bed may introduce occupied or failed nodes into the experimental configuration, resulting in the distortion or even failure of the whole experiment [9].

3. Experimental Bed System Architecture

The whole experimental bed system adopts the three-tier architecture of server, cluster manager, and sensor node for research and design. The design goal of node scale is about 130 and it is dispersed into 15 cluster managers in a grid way to form a mesh network with more than 6 hops. The experimental codes such as algorithms and protocols are run on the nodes. The server and cluster manager cooperate to realize the management, configuration, and control and the whole system is controlled by the server for diagnosis, scheduling, experimental data collection, analysis, and visualization. The system architecture is shown in Figure 1.

In order to realize cluster node programming, diagnosis, and experimental data collection, the cluster manager is undertaken by the computer deploying Linux system to realize cluster autonomy. The server provides a unified, friendly, and highly integrated web experiment portal, and all software components rely on the server task state machine for synchronization and operation. The server is running on a Dell XPS 8900 with 6th Generation Intel i7-6700, 16 GB DDR4 memory and 1 TB 7200 RPM hard drive. It is configured with two network interface cards: one is for the testbed intranet connected with the cluster center laptops and the other one is for the public Internet to access testbed. Sensor nodes adopt two types, low resource with 8-bit processor and subhigh resource with 16-bit processor, and run TinyOS and Contiki systems [10], respectively, to support heterogeneous characteristics. The node topology is shown in Figure 2, and the node hardware is shown in Figure 3.

In the channel setting, due to the fact that IEEE802.15.4 standard is the common key frequency band of WSN and can be used in ZigBee, Wireless HART, 6LoWPAN, ISA100, and other application fields, the sensing node of the experimental bed is configured with 802.15.4 frequency band, and 802.11 frequency band is used for cooperation and data transmission between cluster manager and server, so that the experimental channel and management data transmission channel are separated to avoid interference.

3.1. Design of Underlying Sensor Network. The underlying sensor network consists of 15 clusters with 130 nodes, and the cluster management equipment is undertaken by the notebook computer. The distance between nodes is about 3 feet, which is deployed on the wooden platform. The whole network structure is grid-like, which can form a network forwarding path of more than 6 hops in space.

Because the design of the sensing layer in this paper is based on the limitations of existing hardware and environmental conditions, 130 nodes are used for topology planning according to the space size, but the number of nodes is not limited to this number. The reference can increase or decrease the number according to their own situation, and the system software has no restrictions on this number.

The protocol stack of sensor nodes runs on TI chip CC2530, which is a real system-on-chip solution for ZigBee and intelligent energy applications of IEEE802.15.4. CC2530 also integrates a fully integrated high-performance RF transceiver with 8051 MCU, 8 KB ram, 256 KB flash memory, and strong peripheral support. The sensor module of the sensor node includes various sensors sensing environmental data, and the energy supply adopts USB 5V power supply mode.

The hardware circuit structure of the node is shown in Figure 4.

3.2. Server Structure Design. The C/S mode is adopted between the server and the cluster manager for high robust overall framework planning, a friendly unified web portal is realized on the server side, and a variety of tools are deployed to facilitate the sharing of laboratory bed resources by a large number of users. The key components of the server include web integration interface, database, experimental configurator, experimental controller, and experimental process parameter injector. The key components of the cluster manager include experimental task subcontroller, node programmer, and node diagnostics. The whole server architecture is designed and implemented based on Lamp (Linux Apache MySQL PHP).

Lamp website architecture is an internationally popular web framework, which includes Linux operating system, Apache Web server, MySQL database, Perl, PHP, or Python programming languages [11]. All the products are open-source software, which is an internationally mature architecture framework. Many popular applications adopt this architecture, which is compared with Java/J2EE architecture; Lamp has the characteristics of rich web resources, light weight, and rapid development net architecture and has the advantages of universality, cross platform, high performance, and low price [12]. Therefore, Lamp is the preferred platform for this research design in terms of performance, quality, and price.

The server provides a highly integrated and friendly web experiment portal, supports experiment configuration, management, process control, data collection, analysis, and visualization, and integrates functions such as node real-time diagnosis and experiment process intervention control. The specific interface is shown in Figure 5.

4. System Control Software Design

The software architecture is designed according to the three-tier architecture of the system and realizes the distributed cooperation between layers. In the design of control mechanism, a good control mechanism is studied and designed to ensure the robust, efficient, and reliable operation of the experimental bed system. The main components include experimental planning, experimental resource screening, experimental data export, and experimental bed health status diagnosis. The key design points include robust task state machine, fault-tolerant recovery mechanism, sensor array real-time diagnosis, resource planning, and experimental task scheduling. The structure of control software is shown in Figure 6.

Figure 6 shows the system software architecture, in which different components run at their corresponding layers to streamline the interaction between server, laptops, motes, and users. The web portal provides the users an entrance to schedule and monitor the jobs, which are served by the experiment scheduler and job controller components. The injector service creates a chance to download some

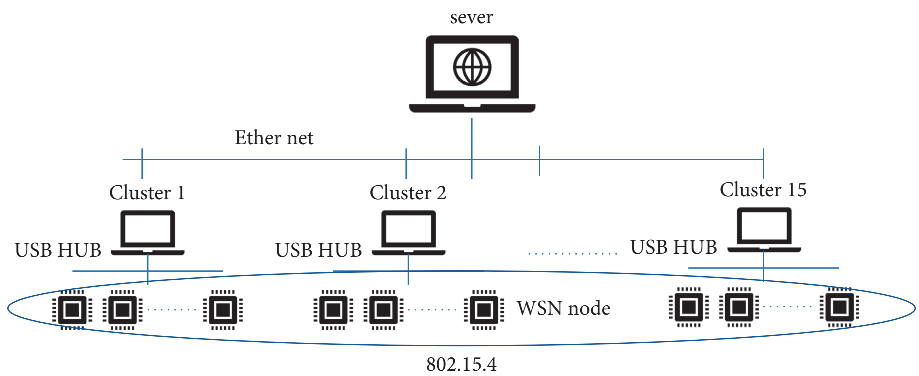


Figure 1: System architecture.

Figure 2: Node topology.

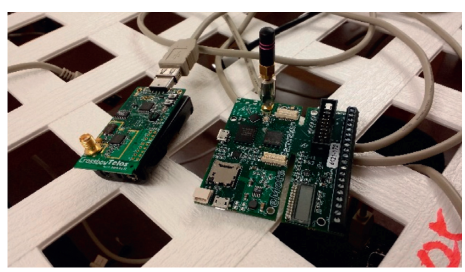


Figure 3: Node hardware of TinyOS and Contiki.

experimental configuration to the specific motes at runtime. The MySQL database on the server is responsible for recording the status of jobs and motes, while the logger FTP server works for the preservation of experimental result files. The subordinate controller running on laptop cooperates with the job controller of server to maintain the state machine. The data & programming logger is to program the motes and use USB ports to collect runtime output data from motes. The doctor is to diagnose node health and update the server database. Some key components will be described in more detail below from the viewpoints of robust and open experimentation.

4.1. Control State Machine. The design of the control state machine is based on the American NetEye experimental bed system. It is a small- and medium-sized control system with good robustness [13].

This state machine is maintained by the server and the distributed cluster manager. As shown in Figure 7, the role of cluster manager is assumed by the laptop. In this figure, the white circle state indicates that the task state is maintained by the server, while the gray circle state indicates that it is updated by the computer acting as the cluster manager. These states reflect the state of the sensor node in real time. User actions can also trigger the state transition. For example, if the user wants to initiate the cancellation of the current task at runtime, all cluster managers will trigger subordinate nodes to stop the current task and switch to the cancelled state. If all task nodes have been switched to the cancelled state, the server will mark the task state as cancelled. In addition, in order to automatically schedule tasks, the system designs a timer event driven task state switching to start the task at a specific time and keep it running within a predefined duration.

4.2. Key Component Design

4.2.1. Runtime Health Diagnosis. The system is designed to execute node health diagnosis at the mote programming stage instead of an exclusive daemon. When the job is

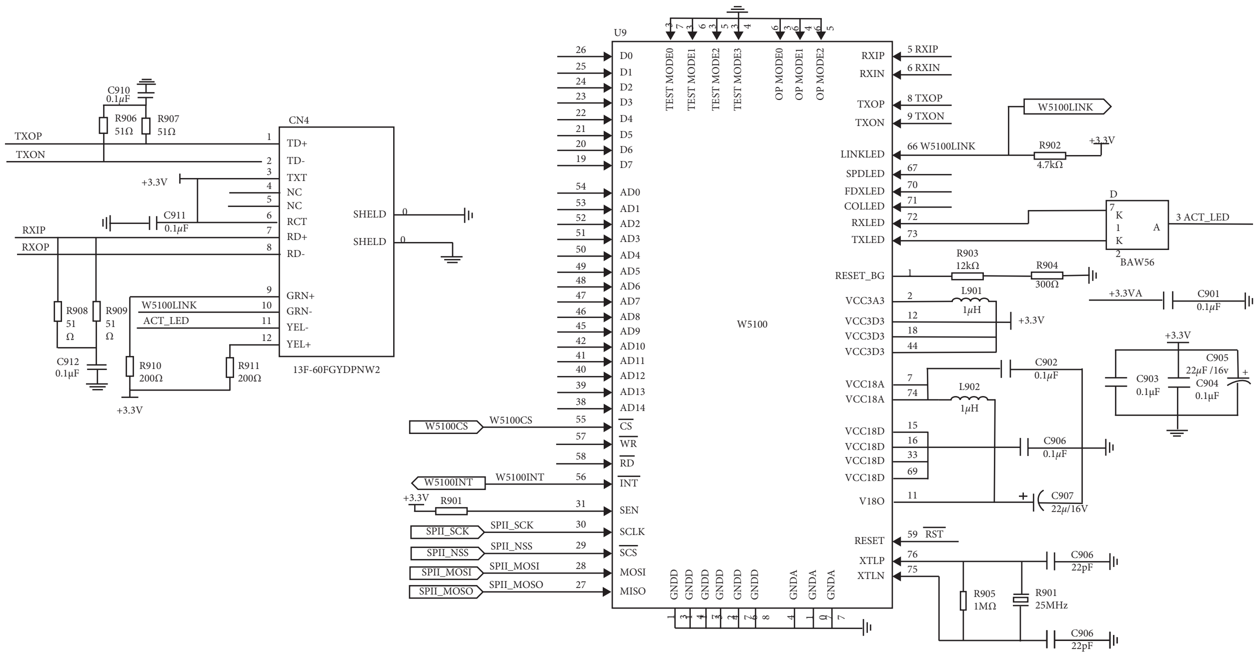
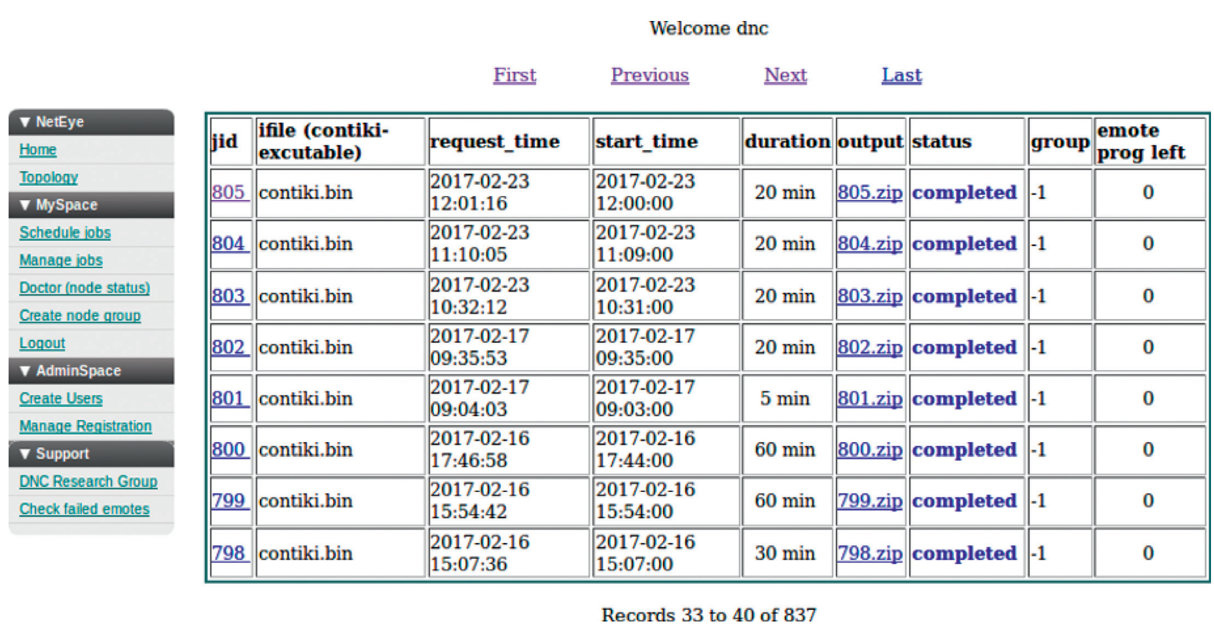


Figure 4: Hardware circuit structure of the node.



jid	ifile (contiki-excutable)	request_time	start_time	duration	output	status	group	emote prog left
805	contiki.bin	2017-02-23 12:01:16	2017-02-23 12:00:00	20 min	805.zip	completed	-1	0
804	contiki.bin	2017-02-23 11:10:05	2017-02-23 11:09:00	20 min	804.zip	completed	-1	0
803	contiki.bin	2017-02-23 10:32:12	2017-02-23 10:31:00	20 min	803.zip	completed	-1	0
802	contiki.bin	2017-02-17 09:35:53	2017-02-17 09:35:00	20 min	802.zip	completed	-1	0
801	contiki.bin	2017-02-17 09:04:03	2017-02-17 09:03:00	5 min	801.zip	completed	-1	0
800	contiki.bin	2017-02-16 17:46:58	2017-02-16 17:44:00	60 min	800.zip	completed	-1	0
799	contiki.bin	2017-02-16 15:54:42	2017-02-16 15:54:00	60 min	799.zip	completed	-1	0
798	contiki.bin	2017-02-16 15:07:36	2017-02-16 15:07:00	30 min	798.zip	completed	-1	0

Figure 5: Server management interface design.

scheduled to run, the diagnosis service is executed with the mote programming. The node status of healthy or not is received by diagnosis service based on the programming success or not. The programming tool will report the error contents including USB port unreachable, flash writing failure, or failure to reboot embedded system. The experimental log will be updated and node status will be refreshed on the page after the diagnosis service received this real-time feedback information. Based on the diagnosis result, the user can decide to continue or cancel this current job to avoid getting an unreliable experimental output. With this strategy, the diagnosis service will not occupy exclusive testbed resources and avoid the competition with experimental jobs.

4.2.2. Automatic Substitution of Failed Nodes. A popular issue during experiment is that the topology may be changed if some motes of this scheduled job fail at the job's beginning [14]. It will cause unreliable experimental output data and mislead the experimental regression and data analysis if the user fails to pay attention to the failed nodes and continue

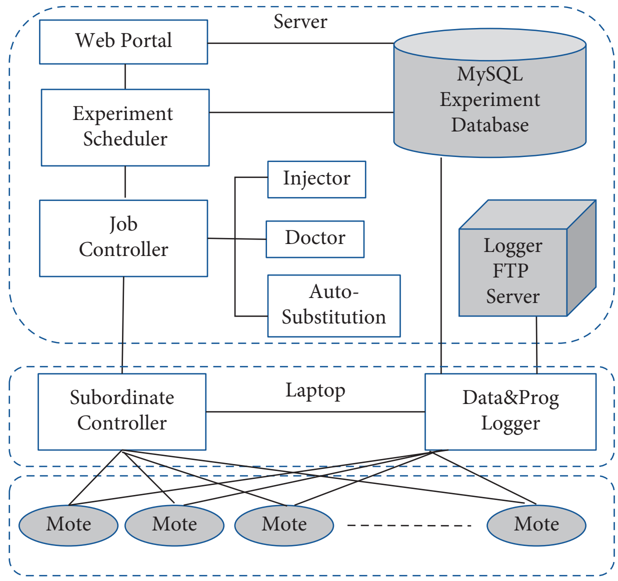


Figure 6: Structure of control software.

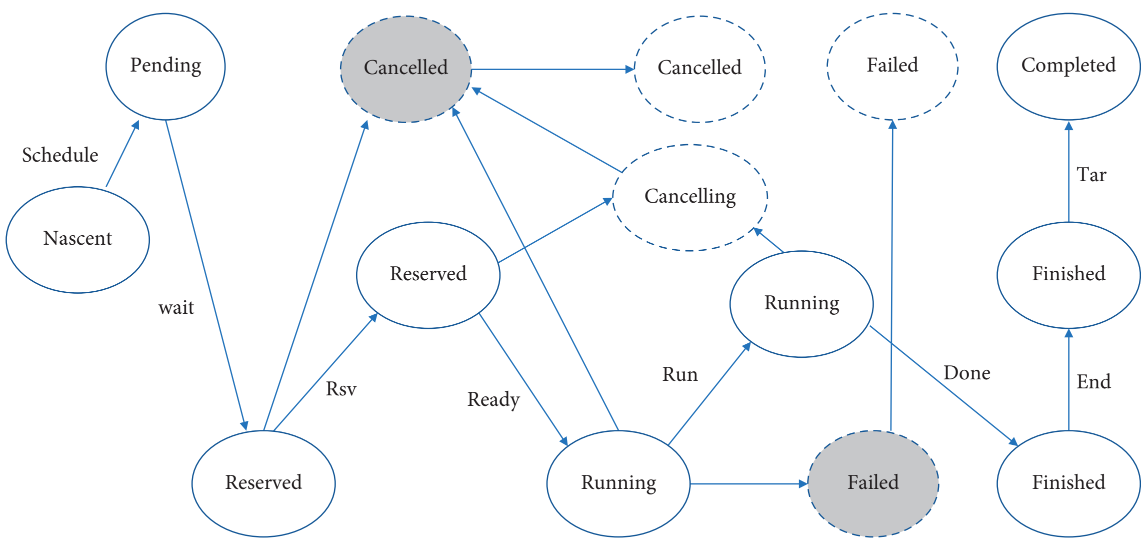


Figure 7: Software state machine.

the experiment [15]. To address this problem and reduce the experimental distortion to a minimum degree, this system provides a strategy to automatically substitute the failed node with a healthy neighbor node. If a mote is diagnosed to be a failure node and reported to server, this system will search and program a healthy neighbor mote to replace it with the minimum change of experimental topology and minimum result influence. All these actions are executed automatically and reported to users.

4.2.3. Reliable and Fast Data and Programming Logger. The role of data logger component is to log experiment data and upload logs to server. It will take a bit long time to program all nodes if an experiment includes a large number of motes, which may delay the experiment [16]. To speed up the programming process, a parallel programming service is utilized in this system. In this system, the comparison experiment of multithreaded and single-threaded node program burning time is carried out, and the results show that parallel programming time of 130 motes can be shortened to 40 s compared to 340 s spent on 10 motes without multithread programming method. In addition, another challenge for data logging is that job with a long-term running time will produce a huge data output to be logged, which may cause the competition of limited TCP connection to upload data files. To solve the transmission failures caused by competition, a TDMA mechanism is used in this system to upload data files, which will avoid producing incomplete data for experimental analysis.

The TDMA mechanism is a time slicing mechanism in the process of experimental data collection and upload. It allocates an average of 500 ms time slice to each node participating in the experiment for rotational transmission, so as to ensure that all nodes can reliably upload data to the server and avoid data loss due to transmission interruption caused by bandwidth competition, resulting in experimental distortion.

4.3. Reliability Strategy Mechanism

4.3.1. High-Integrated Web Interface. The web-based pages developed with PHP and JavaScript serve as experimentation web portal for job configuring, scheduling, and data collection. All tools are seamlessly integrated with the web pages to fulfill their functions. Users can be conducted by the web pages to configure, schedule, monitor, and control their jobs in the whole life-cycle of the jobs. All the stages of job execution can call the service of corresponding tool without causing any inconvenience to users.

The tool of mote health diagnosis integrated in the Doctor page can check the mote status in background and show the mote status such as "busy," "free," and "unavailable." It will help users to decide which group motes should be included in the job and monitor the status during the experiment. Users can also check or download the output data after this testbed finishes the experiment, which will be shown as "completed" on the job list page and a hyperlink will be provided too.

The diagnosis service running as a background daemon can check whether the mote is programed successfully or not and stamp the mote with an unreachable flag if programming fails. As a result, the mote figure on the job schedule page will be masked to disable any other user to include it in his experiment.

4.3.2. Fast State Machine Transition. The system speeds up the state transition during the life-cycle of scheduled jobs to shorten the waiting time of some stages, such as employing parallel daemons in programming and runtime diagnosis. It will improve the experimental efficiency and allow a large number of users to access and share the testbed to carry out their experiments at the same time.

After the experimental test, the switching time of four experimental states is counted by using the experimental scale of 40 nodes, respectively, and the state switching time of the same type is analyzed and compared by using other experimental systems, as shown in Table 1; efficiency increased by about 20%.

4.3.3. Security Locks of Shared Nodes. A competition may result if some motes are scheduled in more than one job and the execution time windows of these jobs are overlapping. As a result of the competition, the jobs may occupy some of the shared motes separately. It will cause the deadlock of these jobs, so the states of these jobs cannot transit correctly as none of these jobs can reserve all the needed motes for the jobs, respectively. Other users also cannot execute their jobs including any of the shared motes because these motes cannot be freed by other jobs trapped in the deadlock. To solve this potential problem, eNetEye locks these shared motes and gives the right to the job with the earliest start time, and the other potential competitive jobs will be delayed for a random factor (between 0 and 1) times its scheduled duration. With such a strategy, eNetEye testbed can avoid the potential risk of job deadlock and improve the facility efficiency.

5. Experimental Analysis

5.1. Wireless Channel Planning Experiment. The sensor node hardware adopts TI CC2530 chip, and its 802.15.4 channel coexists with the data transmission channel 802.11. Since the 15–24 channels of 802.15.4 overlap with the 1–11 channels of 802.11, it is necessary to conduct interference test on other channels of 802.15.4 to screen out the lowest interference experimental channel.

In order to further reduce the interference, the distance based frequency selection algorithm is used for verification, where all wireless nodes are located on the two-dimensional plane; that is, $N_i \in R^2$.

Frequency band belongs to

$$\beta = [f_{\min}, f_{\max}]. \tag{1}$$

The channel bandwidth is f_w, when center frequency f_i belongs to the following scope:

$$f_i \in \left[f_{\min} + \frac{f_w}{2}, f_{\max} - \frac{f_w}{2}\right]. \tag{2}$$

Assign a channel to the sensor N_i node with f_i, and the coordinate of the new node is (U_i, f_i). The model loss function is

$$F_L = \|V - U_i\|_{\rho^2} + \beta\|f - f_i\|_{\rho^2}. \tag{3}$$

Adjust β so as to change the compromise value of frequency and distance, so as to find a subset and get the optimal solution of the following objective function:

$$\varnothing(\{A_i\}, \{f_i\}) = \sum i \int A_i\left(\|V - U_i\|_{\rho^2} + f - f_i\right)^2_{\rho^2}. \tag{4}$$

Since the location of the assigned node is fixed, the variable has only one frequency, where $\varnothing(\{A_i\}, \{f_i\})$ represents the influence of the node on the total unit, for a given node $\varnothing(\{A_i\}, \{f_i\})$ with the smallest value, and the interference is the smallest at this time. We can get the optimization formula in the figure below [17].

$$\varnothing(\{A_i\}, \{f_i\}) = \int A_i\left[\sum i\left(\|V - U_i\|_{\rho^2}^{2} + f - f_{i\,\rho^2}^{\;2}\right)^{\tau}\right]^{\tau}. \tag{5}$$

According to the above results, the signal autocorrelation coefficient is calculated, and Figure 8 is obtained, which shows that the signal can be transmitted and received correctly.

5.2. Topology Control Experiment. In order to realize the simulation of mesh networks with more than 6 hops, it is necessary to set the RF attenuation value and transmission power control level of nodes through experiments, deploy the grid distance from the set nodes, and determine the

Table 1: State switching time comparison.

	Cancelled (ms)	Running (ms)	Failed (ms)	Finished (ms)
Current system	1896	7304	5497	8157
Average reference time	2460	9630	7660	10290

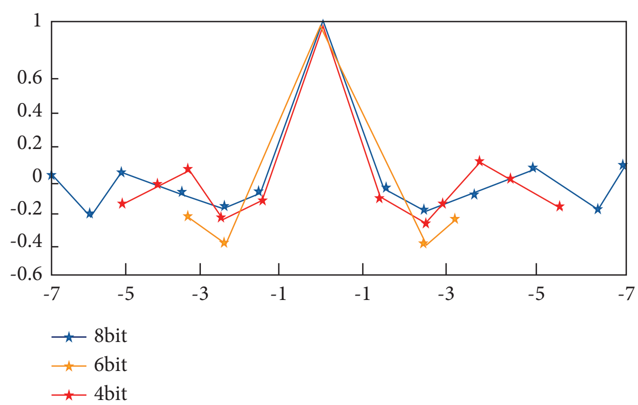


Figure 8: Autocorrelation coefficients of symbols with different lengths.

attenuator parameters and power level by testing and counting the curve relationship between PDR packet forwarding rate and distance.

In the design, the coplanar waveguide (CPW) is used as the basic transmission line, the power is absorbed and attenuated through the T-shaped attenuation circuit, and the circuit is matched with the bypass capacitor to make the circuit maintain stable attenuation in the whole bandwidth. Between the input and output ports, the switch determines whether the attenuation part is connected to the circuit to control the attenuation.

The use of contact switch has two functions. When the switch is pulled down, it can short-circuit the attenuation resistance and become a simple path. When the switch is disconnected, the pull-down plate will form a capacitance with the pull-down electrode. In the design, the capacitance value is changed by adjusting the size and gap height of the electrode plate to match the reactance value generated by the waveguide. In the attenuation circuit, the transmission line works approximately in the circuit with short-circuit terminal at high frequency under signal; when the electrical length $\theta < 90^\circ$, the transmission line is inductive. Its impedance Z_{in} is expressed as follows:

$$Z_{\text{in}} = j \cdot Z_0 \tan \theta. \tag{6}$$

In the above formula, Z_0 is the voltage current ratio along any direction.

The relationship between the attenuation amount in the circuit, that is, S_{21}, and the conversion loss L is

$$L = |S_{21}|^{-2},$$
$$L = \frac{|2Z_0 + Z|^2}{(2Z_0)^2}. \tag{7}$$

The characteristic impedance value required in the attenuation circuit can be deduced from the attenuation. The impedance value consists of two parts. The main part is the characteristic impedance of the T-type attenuation network, and the other part is the equivalent inductive reactance capacitance reactance value in the overall circuit. The capacitive reactance value can be offset by adjusting the size of the waveguide and the capacitance in the switch. For the T-type attenuation network, the circuit is shown in Figure 9 and the three isolation resistors are

$$R_1 = \frac{1+A}{1-A} R_{in} - R_3,$$
$$R_2 = \frac{1+A}{1-A} R_{\text{out}} - R_3, \tag{8}$$
$$R_3 = \frac{2\sqrt{A R_{\text{in}} R_{\text{out}}}}{1-A}.$$

Figure 10 shows the radio model represented by the relation between packet delivery ratio (PDR) and Signal-to-Interference plus Noise Ratio (SINR) at a receiver, which enables this testbed system to mimic various environments.

5.3. System Reliability Experiment. The experimental system is used to test the reliability of the two routing algorithms SPEED and eSPEED. Figure 11 analyzes the performances of the two algorithms in terms of reliability satisfaction. In this experiment, we compare the proportion of data packets that fail to meet the real-time requirement when the reliability requirement ΔP changes from 0.65 to 0.9 under the delay δ of 0.2 s and 0.15 s. As can be seen from Figure 11, when the delay δ is 0.2 s, with the higher reliability requirements, the proportion of data packets in the eSPEED protocol that fail to meet the real-time requirements is lower, which indicates that the method in eSPEED selects the link with higher quality to forward.

We compare the proportion of data packets of the two methods that do not meet the real-time requirements in different network environments. The error rate shows the probability of packet error in the certain model. The higher the error rate, the worse the link quality and the worse the network environment [17]. It can be seen that the higher the error rate, the higher the proportion of data packets that fail to meet the real-time requirements of the two methods, and the advantages of eSPEED are more obvious, because the reliability of the link is considered in this method, while the delay of the link is only considered in SPEED [18]. This shows that the data obtained by the experimental system areis consistent with the characteristics of the two routing algorithms.

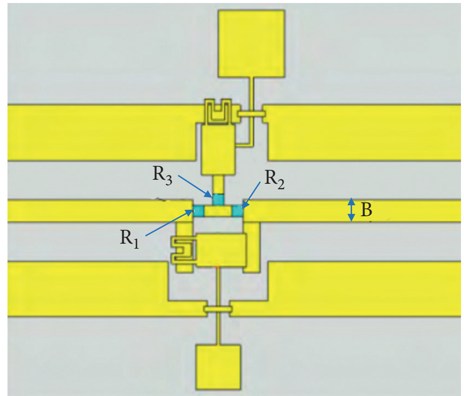


FIGURE 9: Attenuation circuit diagram.

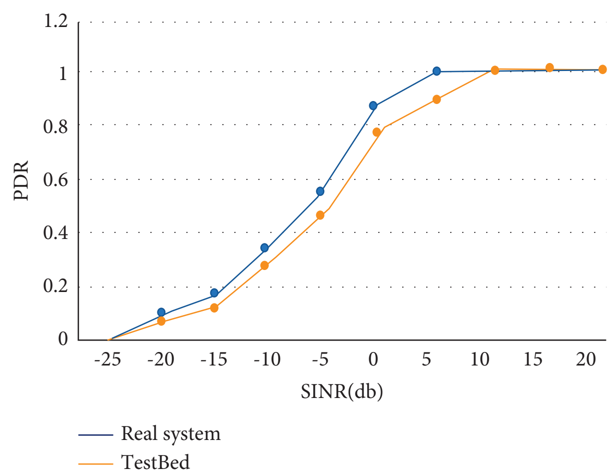


FIGURE 10: PDR versus SINR in system nodes.

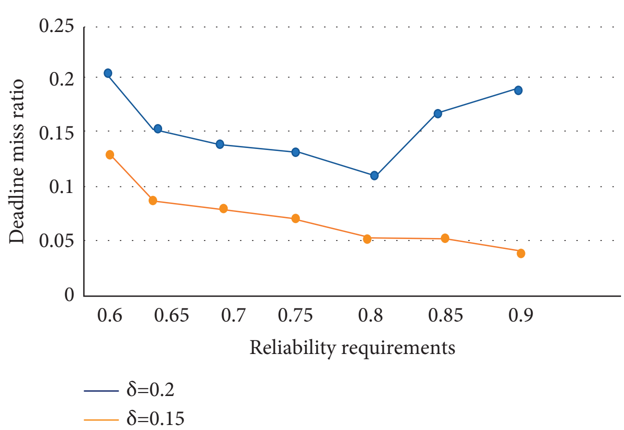


FIGURE 11: Comparison of deadline miss ratio under different reliability requirements.

6. Concluding Remarks

This experimental bed is developed with high fidelity and robustness through mechanisms such as varying environment simulation capability, more on-chip resources, and more restrict runtime synchronization between nodes. Highly integrated web interfaces and other tools are provided to guarantee reliable experimentation. Stable and fault-tolerant state machine combined with runtime health diagnosis and failed node substitution strategy contribute to the robustness of this system. The embedded software is designed using open-source platforms.

The design elements of the experimental system include openness, heterogeneous support, management, resource sharing, experimental control, experimental analysis, and experimental repeatability [19]. Using open-source software for system design is an important principle to maintain openness. For example, TinyOS open-source operating system of Berkeley University has been widely used in WSN experimental bed [20]. In terms of heterogeneity, it can be realized by node fusion of different types, different communication standards, and computing resources. In response to node failure, the management design shall integrate various elements of configuration, operation, safety, and maintenance. Resource sharing can design reservation mechanism under the principle of fairness to divide node subsets to meet the experimental tasks of multiple users. In the design of experimental process elements, topology control, node diagnosis, data acquisition, analysis, and visualization should be supported.

With the rapid development of technology and industry, wireless sensor network technology and other wireless and mobile communication technologies show a trend of continuous integration, such as cross applications with RFID tags, robots, Internet of vehicles, cloud computing, cognitive radio, and content center network (CCN), which continuously excavate and promote the development of wireless sensor networks and optimize and improve their overall performance. This also puts forward higher requirements for the design and implementation of WSN experimental bed infrastructure, such as robustness, heterogeneity, and fault tolerance, so as to provide high-fidelity experimental and regression tools for the research of node datasets, environmental factors, processing algorithms, and network protocols.

Data Availability

The labeled datasets used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Natural Science Foundation of Chongqing (Grant no. cstc2020jcyj-msxmX0196) and Science &Technology Research Program of Chongqing Municipal Education Commission (Grants nos. KJQN202004001 and KJQN202104009), partly funded by Chongqing Technology and Business Institute Project (Grant no. ZZ2020-06).

References

[1] F. De, M. Claudio, L. I. Wei et al., "A systematic review of shared sensor networks," *ACM Computing Surveys*, vol. 48, no. 4, pp. 1–51, 2016.

[2] C. Xia, X. Jin, C. Xu, Y. Wang, and P. Zeng, "Real-time scheduling under heterogeneous routing for industrial Internet of Things," *Computers & Electrical Engineering*, vol. 86, 2020.

[3] X. Ju, H. Zhang, and D. Sakamuri, "eNetEye: a user-centered wireless sensor network testbed for high-fidelity, robust experimentation," *International Journal of Communication Systems*, vol. 25, no. 9, pp. 1213–1229, 2012.

[4] H. Kim, W. K. Hong, J. Yoo, and S. E. Yoo, "Experi-mental research testbeds for large-scale WSNs: a survey from the architectural perspective," *International Journal of Distributed Sensor Networks*, vol. 11, 2015.

[5] D. De, W.-Z. Song, M. Xu, L. Shi, and S. Tan, "Advances in real-world sensor network system," *Advances in Computers*, vol. 90, pp. 1–90, 2013.

[6] TinyOS, "TinyOS," 2012, http://www.tinyos.net/.

[7] F. A. Aderohunmu, D. Brunelli, J. D. Deng, and M. K. Purvis, "A data acquisition protocol for a reactive wireless sensor network monitoring application," *Sensors*, vol. 15, no. 5, Article ID 10221, 54 pages, 2015.

[8] Contiki, "Contiki," 2022, http://www.contiki.net/.

[9] Front, "Frong range observaional network testbed," https://wiki.ucar.edu/display-/iwfr/Front+Range+Observational+Network+Testbed+\%28FRONT\%29.

[10] R. K. Mansano, E. P. Godoy, and A. J. V. Porto, "The benefits of soft sensor and multi-rate control for the implementation of wireless networked control systems," *Sensors*, vol. 14, no. 12, Article ID 24441, 2014.

[11] X. Liu and H. Zhang, "Taming uncertainties in real-time routing for wireless etworked sensing and control," *IEEE Transactions on Automatic Control*, vol. 55, pp. 674–690, 2010.

[12] H. Oh and P. Van Vinh, "Design and implementation of a MAC protocol for timely and reliable delivery of command and data in dynamic wireless sensor networks," *Sensors*, vol. 13, no. 10, Article ID 13228, 2013.

[13] Z. Chen, Z. Han, H. Jiang, Q. Zhu, Y. C. Soh, and L. Xie, "Fusion of WiFi, smartphone sensors and landmarks using the kalman filter for indoor localization," *Sensors*, vol. 15, no. 1, pp. 715–732, 2015.

[14] Lotus, "An advanced wireless node platform," 2012, http://www.memsic.com/support-/documentation/wireless-sensor-networks/category/7datasheets.html?-download=186\%3Alotus.

[15] A. Jiménez-González, Martínez-De Dios, J. Ramiro, and A. Ollero, "An integrated testbed for cooperative perception with heterogeneous mobile and static sensors," *Sensors*, vol. 11, no. 12, Article ID 11516, 2011.

[16] L. Zhang and Y. Guo, "Research on communication interference of Zigbee and WIFI," *Network and Information Engineering*, vol. 17, 2018.

[17] Q. Chen and H. Gao, "The reliability and real-time routing based on uncertain delay in wireless sensor network," *Journal on Communications*, vol. 8, pp. 111–119, 2013.

[18] Q. Chen and H. Gao, "RSPEED：the reliability and real-time routing based on uncertain delay in wireless sensor network," *Journal on Communications*, vol. 34, no. 8, 2013.

[19] M. Sadrishojaei, N. J. Navimipour, M. Reshadi, and M. Hosseinzadeh, "A new preventive routing method based on clustering and location prediction in the mobile Internet of Things," *IEEE Internet of Things*, vol. 8, 2021.

[20] NetEye testbed, "NetEye testbed," 2012, http://neteye.cs.wayne.edu/neteye/home.php.