Review Article

Experimental Research Testbeds for Large-Scale WSNs: A Survey from the Architectural Perspective

Hiecheol Kim, Won-Kee Hong, Joonhyuk Yoo, and Seong-eun Yoo

School of Computer and Communication Engineering, Daegu University, Gyeongsan 712-714, Republic of Korea

Correspondence should be addressed to Seong-eun Yoo; seyoo@daegu.ac.kr

Received 24 May 2014; Revised 18 August 2014; Accepted 5 September 2014

Wireless sensor networks (WSNs) have a significant potential in diverse applications. In contrast to WSNs in a small-scale setting, the real-world adoption of large-scale WSNs is quite slow particularly due to the lack of robustness of protocols at all levels. Upon the demanding need for their experimental verification and evaluation, researchers have developed numerous WSN testbeds. While each individual WSN testbed contributes to the progress with its own unique innovation, still a missing element is an analysis on the overall system architecture and methodologies that can lead to systematic advances. This paper seeks to provide a framework to reason about the evolving WSN testbeds from the architectural perspective. We define three core requirements for WSN testbeds, which are scalability, flexibility, and efficiency. Then, we establish a taxonomy of WSN testbeds that represents the architectural design space by a hierarchy of design domains and associated design approaches. Through a comprehensive literature survey of existing prominent WSN testbeds, we examine their best practices for each design approach in our taxonomy. Finally, we qualitatively evaluate WSN testbeds for their responsiveness to the aforementioned core requirements by assessing the influence by each design approach on the core requirements and suggest future directions of research.

1. Introduction

A typical wireless sensor network (WSN) consists of a large number of small-sized, battery-powered nodes with wireless communications and various sensing capabilities [1]. They are limited in battery energy, CPU power, and communication capability. WSNs attract a great deal of research attention in the past decade [1–7], and in recent years we have seen an increase in prototype WSN deployments to evaluate the feasibility, performance, and cost-effectiveness in a variety of applications, including environment monitoring, infrastructure health diagnosis, and unmanned surveillance [8].

Despite the increase in prototype WSN deployments, the adoption of WSN applications in our real world does not keep the pace. This is phenomenal for large-scaled WSN applications. Of many reasons, the most critical one is the lack of reliability in wireless communication due to unstable protocol behaviors that do not appear in the small-scale setting. For this reason, the complete verification of performance as well as behavior is essential especially for large-scale WSN applications. Traditionally, protocol designers resort to analytic models or simulation to verify and evaluate their proposals [9]. While these methods are relatively simple and convenient, they are inherently limited in the accuracy of evaluation results, as addressed in a number of literatures [10, 11]. To overcome the limitation, especially in the implementation time, comprehensive experiments on a realistic environment are essential. WSN testbeds have long been regarded as a viable option for full-blown physical experiments. Over the years, a great deal of research effort has made considerable progress in building general-purpose WSN testbeds mostly from the research community [12–15]. Aside from these efforts, a large body of works has been devoted to addressing specific issues associated with WSN testbeds such as WSN testbed management [16] and health monitoring [17].

In spite of such decade-long research efforts, researches on WSN testbeds are still ongoing for several reasons. Firstly, the user population of WSN testbeds becomes increasingly diverse, requiring a more distinctive set of features appropriate for new user groups. For example, beyond the academic researchers, users from industries keep increasing
A convenient experimental environment supporting real-time interactions is especially precious for industry users as their works on WSN testbeds are centered largely on code debugging for error-free implementation of WSN networking stacks. Secondly, large-scale WSNs play a key role in IoT (Internet of things) and CPS (cyber physical system) which become increasingly active research areas. Indeed, several prominent testbeds in those fields include a WSN testbed facility as one of their fundamental building blocks [18–20].

WSN testbed researches over years have contributed to the progress of WSN testbeds with their unique innovations. For several reasons, the overall system architecture and methodologies that can lead to systematic advances of WSN testbeds still remain unexplored. Firstly, a well-accepted agreement about WSN testbed architectures does not yet exist because WSN applications are inherently diverse, each with varying requirements. Secondly, decent use cases that can suggest architectural design directions are not available yet. Finally, modest criteria to compare and contrast WSN testbeds are not well established since each individual WSN testbed is based on its own ad hoc design sometimes with an emphasis on a set of specialized features.

Associated with the analysis on overall system architectures and methodologies, several research surveys in literature describe a number of prominent WSN testbeds, highlighting their key features [12–15]. Clearly, they provide a perspective to compare and contrast existing WSN testbeds in terms of physical form-factors, such as hardware infrastructure, software tools, and network size. To the best of our knowledge, they are not sufficient enough to help comprehensively understand an architectural perspective of WSN testbeds; a well-established framework to help analyze architectural features of WSN testbeds is lacking due partly to their focus on specific application domains [12, 13] or on the functional description of WSN testbeds [14, 15].

Our paper, building upon the contributions made by earlier research surveys and complementing their missing part, aims at establishing an architectural framework for analyzing design approaches of the existing WSN testbeds from the architectural viewpoint. We first identify a set of core requirements for WSN testbeds by examining pragmatic experiment scenarios. Subsequently, we establish a taxonomy for WSN testbeds that exposes the design space of a WSN testbed from the architectural viewpoint. Based on the taxonomy, we categorize existing prominent WSN testbeds and examine for each category their designs to get a deep understanding of their approaches and best practices. Based on this understanding, we qualitatively evaluate existing WSN testbeds with respect to the aforementioned core requirements. Finally, we discuss current trends surrounding WSN testbeds and their implications to future WSN testbed architectures.

The rest of the paper is organized as follows. Section 2 introduces the set of core requirements for WSN testbeds that we identify for pragmatic large-scale WSN experiments. In Section 3, we illustrate a taxonomy for WSN testbeds which exposes the architectural design space of WSN testbeds, thereby enabling classifying WSN testbeds from the architecture perspective. In Section 4, based on our taxonomy, we provide a comprehensive survey on design practices of existing prominent WSN testbeds. In Section 5, we evaluate WSN testbeds with respect to the core requirements identified in Section 2. Finally, in Section 6, concluding remarks and future researches are offered.

2. Requirements for WSN Testbeds

Recent advanced WSN testbeds are intended for public use by external researchers and developers from various application domains. If a WSN testbed is to be useful for them, it has to address numerous requirements. This section discusses important requirements for general-purpose WSN testbeds. In Section 2.1, we introduce a generic organization of WSN testbeds and its associated terminology in order to set the context of the discussion. In Section 2.2, we illustrate the set of core requirements that we identify from the pragmatic perspective.

2.1. Generic WSN Testbed Organization.

Apparently, it is not easy to bring a typical organization for WSN testbeds since each one is built with its own unique design to address a set of specific requirements of interest. To frame a context for discussion, we thus identify the set of generic building blocks of a WSN testbed through a close examination of the experiment lifecycle. In general, an experiment on a WSN testbed consists of a series of steps, crudely experiment setup, experiment execution, and result analysis. Following these steps, users need to interact with the WSN testbed facility. The first essential interaction is to reserve a set of nodes from a WSN testbed. Subsequently, users have to load embedded software into those reserved nodes, that is, reprogramming. Another essential interaction is to monitor the experiment state. For evaluating distributed sensing applications, continuous data logging from each individual node is essential. The log data enables users to obtain full insight into the operation of the WSN application and further allows debugging or analyzing the behavior or the performance as well. Finally, users are essentially engaged in sporadic control activities such as starting nodes for initiating an experiment execution.

The aforementioned essential user interactions expose the least set of building blocks that a WSN testbed has to be equipped with. The first one is a dedicated network infrastructure, called usually back-channel, for handling management traffic between nodes and management computers which are produced for user interactions including monitoring, control, and reprogramming. Using the native radio channel of the WSN for the experimental traffic can potentially disturb the behaviors of WSN communication, eventually distorting evaluation results. The second one is a computing infrastructure such as repositories for log data and servers for hosting management services. In this paper, we call the computing infrastructure a testbed management server or a testbed server in short.

Figure 1 depicts the generic view on the organization of WSN testbeds based on the above discussion. A WSN testbed consists of three parts, a WSN field, a back-channel infrastructure, and a testbed server.
2.1. WSN Field. A WSN field is a physical WSN to host WSN applications under experiments. The principal element of a WSN field is a collection of sensor nodes. From now on, a sensor node will be called usually a node in short. Across the existing WSN testbeds, WSN fields exhibit several differences. The first difference is the node hardware. A number of WSN testbeds employ the sensor node platforms commonly used by the WSN research community. For instance, Telos [21] and Tmote series [22] belong to them. In several WSN testbeds, custom-designed nodes are used for their WSN testbeds. These nodes usually have additional features associated with WSN testbeds such as back-channel communication interface and power profiling hardware. Other differences include, but are not limited to, an apparatus for supporting mobility and power provision schemes.

2.1.2. Back-Channel. A back-channel is the network to carry the management traffic. Typically, sensor nodes do not support a back-channel functionality and thus the infrastructural support is provided independently of the WSN field. Across the existing WSN testbeds, designs of their back-channels exhibit a large variance in terms of communication methods and hardware organizations. Some WSN testbeds use onboard wired communication ports such as RS-232c and USB port. Others often adopt broad-band networks with some additional network devices such as back-channel gateways. In implementing a back-channel, each WSN testbed makes its own trade-off with respect to such criteria as implementation cost, bandwidth, and scalability.

2.1.3. Testbed Server. A testbed server is a computing facility for experimentation and administration services, respectively, for experimenters and administrators. A testbed server consists of both hardware and software. A testbed server exposes the capability of a WSN testbed to the outsides and is thus regarded as an access point for users. Services provided by a testbed server will be called testbed services, and the software, which implements those services, is called testbed software. Sometimes, testbed software is disintegrated into a server part and a client part as in the general client/server model. In this case, they are, respectively, called server-side software and client-side software. For users to access testbed services, the web interface is used in many WSN testbeds, while custom-tailored or publicly distributed client-side software is used in several other WSN testbeds.

2.2. Requirements. The embedded and distributed nature of WSNs, potentially large-scale WSNs, and diverse WSN application domains render the design of a versatile WSN testbed inherently difficult. The design should reflect perspectives of diverse entities including the WSN testbed infrastructure, WSN applications, and WSN testbed users. With this in mind, we select three core requirements for WSN testbeds: scalability, flexibility, and efficiency.

2.2.1. Scalability. This means the capability to increase the number of nodes in the WSN testbed without restricting the robustness in experiments. One of important goals for building a WSN testbed is to evaluate large-scale WSN applications since the evaluation of small-scale WSN applications can be undertaken just with some ad hoc apparatus. A typical WSN application potentially is prone to clash when running on tens or hundreds of nodes, even though it is sufficiently verified in a small-scale setting. For instance, unexpected network behaviors may appear due to increased topology changes. Software bugs that are not easily discovered in a small-scale setting such as a bounded-buffer overflow may occur. Scalability is thus important for putting scenarios of large-scale real-world deployments into experimentation.

2.2.2. Flexibility. This means the capability to allow users to conduct realistic experiments for a wide range of WSN applications. Considering the high deployment cost of a WSN testbed, the flexibility is essential in the WSN testbed design since it ultimately determines the extent of their utilization. For higher flexibility, the design of WSN testbeds has to effectively deal with the high degree of hardware and environmental diversity across the large spectrum of WSN applications. As for the hardware diversity, it is desired to flexibly replace nodes or at least radio and sensor modules in a WSN testbed appropriate to the hardware requirement of the target application. On the other hand, as for the environment diversity, a WSN testbed should allow users to flexibly set up experimental environments appropriate to their WSN applications. Environmental factors subject to considerations include RF characteristics, such as radio propagation range and interference, and node dynamics like physical node failure and node mobility.

2.2.3. Efficiency. This means the capability to enable users to conduct experiments effectively in terms of time and labor. Ordinary users do not have deep understanding on the WSN testbed itself. Moreover, as previously discussed, a typical WSN experiment is inherently complex due to the embedded and distributed nature of WSNs; it requires...
3. Taxonomy

The large spectrum of WSN applications results in numerous WSN testbeds distinctive in many design features. For systematic advances of WSN testbeds, it is very important to be able to sort out these features and then characterize their designs coherently from an architectural perspective. To this end, we propose a taxonomy. In building it, we first explore a set of design domains at the level of architectural building blocks. For each design domain, we identify a set of disjoint design approaches in its design space. Figure 2 shows our taxonomy that consists of four design domains: back-channel organization, service architecture, WSN field structure, and WSN layout. In the figure, design approaches are also listed, respectively, for each design domain. In this section, our taxonomy is explained with each core design domain in a separate subsection.

3.1. Back-Channel Organization. The back-channel organization of a WSN testbed, which is the first design domain in our taxonomy, is important because it affects many aspects of the WSN testbed which include cost-effectiveness, physical form factor, and scalability. A number of approaches have been employed in the design of a back-channel. We classify their organizations into three: native-BC (back-channel), broadband-BC, and hybrid-BC, as shown in Figure 3.

3.1.1. Native-BC. A back-channel is implemented by using on-board serial communication. Except for a few WSN testbeds using RS-232c, the USB communication is typically used in WSN testbeds. Each serial port is wired directly to the testbed server. In case of USB, indirect wiring via USB hubs is used as well.

3.1.2. Broadband-BC. Broadband networks, mostly either Ethernet or WiFi, are used to implement back-channels. Each node in the WSN testbed is equipped with a dedicated device for protocol conversion. While the device is called differently such as a sensor host [23], a super node [24], or just a gateway [25], we call it back-channel gateway (BGW) as it provides the protocol conversion between on-board serial communication, which are mostly RS232c, and the broadband network. According to the way to implement the
back-channel gateway, we divide this broadband-BC further into two approaches. One is to use an adaptor module to be piggy-backed on the node and the other is to use an external controller which is usually a single board computer. They are called, respectively, broadband-BC with an on-board BGW and broadband-BC with an external BGW, being separately dealt in the taxonomy due to their distinctive influence on the overall WSN testbed architecture. As it is named, the key advantage of the broadband-BC is to get higher bandwidth over the native-BC. Moreover, this approach enables exploiting standard networking equipment, facilitating the implementation of a back-channel.

3.1.3. Hybrid-BC. A combination of a native-BC and a broadband-BC is used to implement a back-channel. The key motivation is to reduce the number of back-channel gateways by dividing nodes in the WSN field into a set of groups, where each group is managed by a single back-channel gateway. Hence, the network between the nodes in a group and their back-channel gateway is in a form of the native-BC, while the network between back-channel gateways and the testbed server is in a form of the broadband-BC, as shown in Figure 3.

3.2. Service Architecture. Over the hardware resources of a WSN testbed, a set of services is to be provided for users who are experimenters, administrators, or technical managers. The organization for these services, which we will call service architecture, is important in the WSN testbed design. For instance, it affects the freedom to customize the client part of testbed software. Hence, we select the service architecture as the second design domain in our taxonomy. Associated with the service architecture, we identify three disjoint design approaches: monolithic, tightly-coupled, and loosely-coupled, as shown in Figure 4.

3.2.1. Monolithic. This architecture assumes the hardware infrastructure where a single computer serves as both a client desktop and a testbed server. The testbed software is not disintegrated into a client part and a server part and thus is implemented monolithically into a single piece.

3.2.2. Tightly Coupled. This service architecture assumes a WSN testbed that is tiered into three layers: clients, a testbed server, and a WSN field. A client desktop computer and a testbed server host client-side software and server-side software, respectively. These are tightly coupled such that the mission of client-side software is solely to deliver the services provided by server-side software to users; client-side software is not permitted to provide its own proprietary services. In this service architecture, the whole testbed services are usually implemented centrally as server-side software, and client-side software provides only the user interface to access those services.

3.2.3. Loosely Coupled. Different from the aforementioned tightly coupled one, this service architecture allows client-side software to provide its own proprietary services independent of server-side software. In other words, client-side software not only serves the delivery mission, but also can be extended to provide a set of functionalities demanded by experimenters. In this architecture, client-side software is often allowed to directly interact with nodes in the WSN field, as depicted by a line in Figure 4(c).

3.3. WSN Field Structure. Across the WSN testbeds, WSN fields are different not only in their node hardware, but also in their structures. As the WSN field structure, in particular, significantly influences various important aspects including heterogeneity and scalability, we select it as the third design domain in our taxonomy. In exploring its design approaches, we introduce a term: WSN cluster. A WSN cluster is a section of a WSN field in which communication between nodes is made only via the in-band radio hops; all the nodes reachable only via any gateway are excluded. A WSN testbed can be built with either a single WSN cluster or multiple WSN clusters. A multiclustered WSN testbed is sometimes constructed by federating multiple independent WSN testbeds. Consequently, we divide the design space of the WSN field structure into three design approaches: singular, clustered, and federated.
3.3.1. Singular. A singular WSN testbed refers to a WSN testbed with a single WSN cluster. This is the most popular since experiments to evaluate the scalability of WSN networking protocols, such as routing protocols, are usually concerned with the number of nodes at the WSN cluster level rather than at the entire WSN field level (Figure 5).

3.3.2. Clustered. A clustered WSN testbed refers to a WSN testbed composed of more than one WSN cluster. WSN clusters are connected with gateways and spread over a wide geographical area. Generally, the cluster boundary is transparent to users, and thus all the nodes over the WSN clusters of a WSN testbed are collectively managed by a single management entity.

3.3.3. Federated. A federated WSN testbed refers to a WSN testbed constructed by combining multiple independent WSN testbeds. Although each member WSN testbed is remotely located and independently administrated, the federation provides a common view to users in terms of the administration policy and the way to utilize each individual member WSN testbed. The federation concept is not new. It is conceived by pioneering research initiatives such as GENI [26] and FIRE [27] to construct a large-scale heterogeneous experimental infrastructure for the future Internet.

3.4. WSN Layout. In deploying a WSN testbed, the way to place WSN nodes determines physical form factors such as physical distance between nodes. These factors may affect several important aspects of the WSN testbed including deployment and maintenance cost and the possibility to control node density within a radio propagation range. In current best practices, options for the WSN layout are divided into two approaches, namely, native and grid.

3.4.1. Native. This layout places WSN nodes in a similar way to general WSN deployments. WSN nodes are separated subject to ordinary radio propagation ranges. Typically, distance between nodes ranges from 10 to 30 meters.

3.4.2. Grid. This layout places WSN nodes in a relocatable fixture in the form of a two- or three-dimensional grid. Typically, distance between nodes ranges from a few ten centimeters to a few meters.

4. Designs

Based on our taxonomy discussed in the previous section, this section provides a survey of existing prominent WSN testbeds. In Section 4.1, we classify WSN testbeds according to our taxonomy. In subsequent subsections, we present a detailed examination of their designs, respectively, for each design domain in our taxonomy.

4.1. Classification. The scope of WSN testbeds in our review is devoted to general-purpose research WSN testbeds with focus on protocol-centric experiments in a large-scale setting. Consequently, numerous WSN testbeds are excluded from our review. Firstly, research WSN testbeds without back-channel support are excluded since the absence of back-channel potentially causes the degradation of fidelity in their evaluation results. Secondly, trial WSN testbeds are excluded. Here, a trial WSN testbed refers to a WSN testbed that is deployed in a real geographical area for evaluating the feasibility of a WSN application in a particular domain before its full-blown real deployment. Trial WSN testbeds are usually specialized to a set of domain-specific requirements; therefore, it is not adequate to evaluate them from the general-purpose viewpoint. Finally, IoT-centric WSN testbeds are also excluded. A heterogeneous mix of IoT devices, such as RFID and domain sensors, is considered out of the scope of this paper. Nevertheless, among the aforementioned WSN testbeds, a number of them embody prominent features. For instance, SmartSantander [20] provides a city-scale experimental facility with 20,000 sensors in support of typical applications and services for a smart city, allowing horizontal and vertical federation.

Table 1 provides a featured summary of the research WSN testbeds examined in this paper. We try to select a comprehensive collection of prominent research WSN testbeds to the best of our knowledge, but we must acknowledge that
| WSN testbed       | Nodes/deployment                                                                 | Features                                                                                                                                                                                                 |
|------------------|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ASSERT [28]      | 40 nodes including 25 Mica2, 1 laboratory office                                | Flexible control of radio environment based on specialized hardware, controllable network topology, supporting multiple nodes based on the UUT concept, and Linux-based specialized controller for back-channel                     |
| CPN [29]         | 92 nodes (TelosB, IRIS, and MICA), over different buildings                     | Back-channel based on UTP-to-USB, persistent device naming, and powerful visualization tool support                                                                                                |
| EasyTest [30]    | 16 EZ521 nodes, 1 laboratory office                                             | Multiradio support based on custom-designed nodes and support 802.11 b/g experiments based on EZ271 nodes                                                                                             |
| Emulab [31, 32]  | 31 nodes (25 Micaz, 6 mobile), 1 laboratory office                             | First WSN testbed with remote access capability, support for mobility with Garcia robot, and MIB600CA-based back-channel                                                                            |
| Indriya [33]     | 127 TelosB nodes, 3D deployment over 3 floors                                   | Active USB infrastructure for back-channel and adopting the Motelab management software                                                                                                                  |
| Kansei [34]      | 210 XSM and Tehran sky node, 1 laboratory office                               | One of pioneering WSN testbeds, mixed-simulation support, mobility support using robots, event injection capability, and deployed on grid fixtures                                                   |
| KanseiGenie [35] | 576 nodes, 1 laboratory office                                                   | Federated WSN testbed based on GENI environment, deployed with Kansei and NetEye, and high node heterogeneity (96 XSM, 384 TelosB, and 96 iMote2)                                              |
| KonTest [36]     | 60 TelosB nodes, 6 office rooms                                                  | One control PC per room directly connected with nodes with USB back-channel                                                                                                                |
| LabVIEW [37]    | 23 MicaZ and 8 cricket nodes, laboratory office                                | PC-based infrastructure, real-time visualization tools based on popular LabVIEW programming environment                                                                                             |
| Motelab [38]     | 190 TelosB notes, 3 floors of a laboratory building                            | First WSN testbed with web-based services, in situ power measurement, usage of Stargate Netbridge for back-channel, and deployed on grid fixtures                                               |
| NetEye [39]      | 130 TelosB nodes, 1 laboratory office                                           | Static 3db attenuator to each antenna for different power levels, usage of laptop computers for back-channel                                                                                        |
| Sensei-uu [23]   | Mostly TelosB nodes                                                             | Supporting smart phone and Lego mobile node, supporting mobility, usage of WL-500G premium broadband routers for back-channel gateways                                                      |
| SensLAB [40]     | 1024 WSN430 nodes, 4 different sites in France                                  | MAC heterogeneity (521 with 802.15.4 MAC, 512 with free MAC layer), radio monitoring based on control nodes, energy measurement, repeatable mobility via artificial train, and VPN connectivity between sites |
| IoT-LAB [41]     | 2700 nodes, 6 sites in France                                                   | Federated architecture with OAR-based reservation support, a successor project of SensLab, highly heterogeneous nodes (MSP430, STM32, Cortex A8-based nodes), and multiple OS support (FreeRTOS, Contiki, Riot, TinyOS, and OpenWSN Linux) |
| SignetLab [42]   | 48 eyesIFx-v2 nodes, 1 laboratory office                                        | Back-channel based on the active USB infrastructure, services based on a single control PC, and Java-based client-side software                                                                   |
| TWIST [24]       | 204 nodes (102 TelosB, 102 eyesIFX), 3 floors in an office building             | Flat, hierarchical, and segmented architecture, a clustered WSN testbed with 3 sections, and usage of 40 NSLU2-based super nodes for the back-channel                                                      |
| WSNTB [43]       | 34 nodes (Octopus I II), 2 laboratory offices                                   | Specialized hardware to support power consumption and timing measurements and distributed management                                                                                  |
| WeaverLab [44]   | 80 WSN120 nodes, 1 laboratory office                                            | Node heterogeneity based on modular node design, a dense grid of 1.5 m × 2 m, visual user interface with real-time interaction, and loosely coupled service architecture                                           |
| WISEBED [25]     | 750 nodes, indoor and outdoor, 9 different locations in Europe                  | Federated WSN testbed, mixed-simulation support, in situ power measurements on some nodes, mobility support using 40 mobile nodes, high node heterogeneity (200 (Sense, 143 TelosB, 108 G-Node, 100 MSB-A4, 44 SunSPIT, 60 pacemate, and 24 Tnode) |
several notable research WSN testbeds are excluded due to our ignorance.

Figure 6 depicts the classification result of the WSN testbeds according to our taxonomy. For pictorial clarity, only three design domains are depicted. In the figure, the back-channel organization, the WSN field structure, and the service architecture occupy, respectively, one axis in the three-dimensional diagram. The result on the remaining design domain, that is, the WSN layout, is reported in Table 3.

4.2. Back-Channel Organization Domain

4.2.1. Native-BC Organization. This design approach provides a cost-effective and handy way to implement a back-channel. The use of an on-board external communication interface as the back-channel makes any other communication equipment unnecessary, thereby enabling using nodes without any modification. On the contrary, the native-BC organization has inherent limitations associated with both wire-length and the number of physical connectors. As a result, this design approach has been used mainly for small-scale WSN testbeds mostly using desktop computers as their testbed servers.

In practical designs of the native-BC organization, USB is a preferred option over RS232c. This is because USB is more extensible than RS232c in terms of both the number of ports and cabling coverage. The maximum length of an ordinary passive cable is limited to 5 meters. On the other hand, with active USB cables, the coverage can be extended to 25 meters as daisy-chaining is allowed up to five active cables. In terms of ports, a network based on USB hubs can constitute 127 physical ports, each connecting one node [36]. Consequently, a design of the native-BC organization falls on one of three methods: RS232c, a passive USB network, and an active USB network. LabVIEW [37] takes RS232c. Kontest [36] and SignetLab [42] adopt the active USB network method.

4.2.2. Broadband-BC Organization. As discussed earlier, the broadband-BC organization uses a broadband network to implement a back-channel. Most of the WSN testbeds in this design approach have adopted Ethernet or WiFi as their broadband networks just for taking advantage of the commercially mature technology. Therefore, designs of the broadband-BC organization are characterized by the way to implement the back-channel gateway which must be provided for each node. Below, we examine the design of each variant of this approach.

Broadband-BC with an On-Board BGW Module. A back-channel gateway is implemented as a hardware module, which we call a BGW module, and can be piggy-backed directly to a node. Typically, a BGW module uses commercial TCP/IP chip-set and provides a protocol conversion capability such as RS232C-to-TCP/IP or USB-to-WiFi. Numerous WSN testbeds take this design approach including EasiTest [30], Emulab [31], WSNTB [43], and WeaverLab [44]. In some cases, the module is integrated directly in the node hardware. This is taken especially when a new proprietary node is developed for their WSN testbeds. The embedded integration enhances their hardware reliability, and the installation of the WSN testbed becomes less troublesome due to fewer hardware pieces. For example, ASSERT [28] and EasiTest [30] develop their own nodes in which, respectively, a 10/100 Ethernet module and a IEEE 802.11b/g WiFi module are integrated inside their hardware.

Broadband-BC with an External BGW Controller. A back-channel gateway is implemented as an external controller which is attached to a node via RS232c. As for the controller, a single-board computer is usually used. For instance, Kansei [34] uses XSS (Extreme Scale Stargate) single-board computers, and Motelab [38] uses Stargate NetBridge which is similar to XSS in its hardware specification.
4.2.3. Hybrid-BC Organization. The main aim of the hybrid-BC organization is to reduce the number of gateways by eliminating the need for coupling each individual node to a dedicated back-channel gateway, while securing the property of the large-scale WSN testbed deployment of the broadband-BC. Each gateway device thus serves more than one node. A hybrid-BC organization is composed of three parts: native-BC networks, back-channel gateways, and a broadband-BC network. In practice, designs of the hybrid-BC organization are primarily distinguished by the back-channel gateway. Across the existing WSN testbeds, choices of both the native-BC part and the broadband-BC part are almost the same with the pure hybrid-BC approach. As for the native-BC organization, a USB network is used, and Ethernet or WiFi is used for the hybrid-BC part. Indriya [33] and TWIST [24] use an active USB network, while NetEye [39], Sensei-uu [23], and WISEBED [25] use a passive USB network.

Designs of the hybrid-BC organization are always supposed to use external BGW controllers for back-channel gateways. TWIST [24] uses NSLU2 single-board computers. In its deployment, 46 wall-powered NSLU2s are installed to manage 204 WSN testbed nodes. On the contrary, Indriya [33] and NetEye [39] use PCs. In Indria, 6 Mac Mini PCs are used to manage 127 nodes, where each Mac Mini PC is capable of servicing more than 100 USB devices. NetEye uses 12 Dell Vestro laptops for 130 nodes. As an exceptional case, CPN [29] employs a USB-to-UTP converter as a back-channel gateway where each USB-to-UTP converter can serve up to 4 nodes. In this design, Ethernet wires for the broadband-BC part eliminate the need for active USB cables or additional hubs as signal repeaters in its native-BC part.

4.3. Service Architecture Domain

4.3.1. Monolithic Service Architecture. This design approach is the most simplistic among the three design approaches in the service architecture domain. Their services are usually built in an ad hoc fashion. Further, their management functionalities may not be explicit, leaving the details to developers. The simplicity of this service architecture enables a fast prototyping of a working WSN testbed [28, 37]. Also, users can freely bring services suitable for their own experiments since they do not need to be concerned about the very complex issues raised when sharing WSN testbeds. For instance, LabVIEW [37] offers real-time visualization of monitoring data by using the popular LabVIEW programming environment. ASSERT [28] enables user to perform delicate configurations entailed for each individual experiment such as setting radio topology and attenuation.

4.3.2. Tightly Coupled Service Architecture. As discussed earlier, in this design approach, the whole testbed services for users are provided by server-side software, and client-side software has a single role to enable users to access those services. A number of well-established multiuser WSN testbeds take this service architecture, as shown in Figure 6. Due to the clear-cut role of client-side software, the designs of client-side software over these WSN testbeds do not show any significant differences. They usually adopt a web browser to provide only a thin interface for accessing the remote services on the testbed server. However, their design practices of server-side software exhibit a large variance in their software and hardware infrastructures. Simplistically, several WSN testbeds are built upon a single server with basic management services such as accounting, reservation, and data logging [31, 33, 38, 39]. On the other hand, several WSN testbeds, which include Kansei [34], KanseiGenie [35], SensLab [40], TWIST [24], and WSNTB [43], employ a cutting-edge infrastructure with a set of sophisticated services over several dedicated servers. For instance, SensLab [40] implements the server-side software based on state-of-the-art standard technologies including an OSGi framework, hypervisor, LDAP directory tree, SQL database, and Apache Tomcat as its servlet container.

4.3.3. Loosely Coupled Service Architecture. In this design approach, the role of client-side software is not only to provide accesses to server-side software, but also to provide proprietary services for users. Instead of web browsers typically used in the tightly coupled architecture, either customization or a third-party implementation of client-side software can thus be conducted. WSN testbeds adopting this architecture take different emphases and methods in their implementations [23, 25, 44]. For instance, WISEBED [25] aims at creating technology independent client-side software. It provides clear-cut service APIs for the implementation of client-side software. All APIs are specified as standard web services using WSDL documents and a human-readable documentation. As server-side software provides a complete set for testbed services, client-side software is usually slim with only the access capability. On the other hand, WeaverLab [44] is motivated to achieve cost-effectiveness by slimming the management infrastructure as well as to provide users with direct interactions with nodes. Testbed software is split over both client computers and the testbed server. The client-side software can provide users with experiment-related services such as reprogramming, monitoring, and data logging. The server-side software is focused on administration and maintenance functionalities. Decoupling services based on the clear notion of user services and management services offer an enlarged freedom to evolve individually along its own path. As a result, WeaverLab [44] provides a rich set of GUI-based experimental supports within the client-side software, which can be even more specialized on demand from a third party.

4.4. WSN Field Structure Domain

4.4.1. Singular WSN Field. Most of the WSN testbeds take this WSN field structure, as shown in Figure 6. The WSN cluster in this singular structure is organized mainly with homogeneous nodes except for several WSN testbeds including EasiTest [30], TWIST [24], and WSNTB [43]. Compared with clustered or federated WSN testbeds, singular WSN testbeds are thus relatively limited in hardware heterogeneity of WSN testbed nodes. Traditionally, a singular WSN testbed
is managed as a stand-alone facility administrated by a single institution.

4.4.2. Clustered WSN Field. Only a few WSN testbeds take this WSN field structure \[29, 43\]. WSN testbeds adopting this WSN structure are deployed either on single site or on multiple sites. As the former case, WSNTB \[43\] is composed of two WSN clusters within a single site, where WSN clusters are different from each other in the radio communication method. When the radio severance occurs among discrete spaces such as rooms and floors, a clustered WSN field structure is also adopted on a single site. As the latter case, WSN clusters of CPN \[29\] are deployed, respectively, on a separate building and are connected via the Internet. Due to the radio severance between WSN clusters, the scalability in the context of this paper may remain the same for both cases, even if the size of the WSN testbed increases.

4.4.3. Federated WSN Field. Several WSN testbeds are built around the federation concept in pursuit of scalability and heterogeneity. KanseiGenie \[35\] in Ohio State University refactors the Kansei WSN testbed into a federation version using GENI \[26\]. It consists of a site at the Ohio State University, which has four different sensor fabric arrays, and a site at Wayne State University, which has two different sensor fabric arrays. SensLAB \[40\] aims at unifying four discrete heterogeneous WSN testbeds into a single WSN testbed of 1,000 nodes. WISEBED \[25\] demonstrates the virtual WSN testbed that comprises transparently physical WSN testbeds along with simulated and emulated elements. Recently, IoT-LAB \[41\], the successor project of SensLAB, delivers a more robust federated WSN testbed based on advanced federation software technologies.

An eventual challenge of building a federated WSN testbed is to provide users with transparent access to member WSN testbeds which usually comprise heterogeneous resources under multiple independent administrative entities. Because of the challenge, building a federated WSN testbed entails a supplemental network and a software infrastructure over the stand-alone WSN testbed facility. The infrastructure network of a federated WSN testbed consists of intrasite networking and intersite networking. Firstly, intrasite networking is quite similar across the federated WSN testbeds. They are typically based on VLAN (virtual LAN). In particular, WISEBED employs an overlay network in which the site gateway encapsulates and relays data between the serial link of the attached nodes and the site server. Secondly, intersite networking for the interconnection between member WSN testbeds exhibits a large variance across the federated WSN testbeds. WISEBED and KanseiGenie use the public network. Testbed servers of member WSN testbeds are directly connected to the Internet without any specific network facility. In contrast to them, SensLAB interconnects member WSN testbeds with VPN (virtual private network) via a dedicated router in each site.

Software infrastructure for federation is focused on the transparent access to the resources in member WSN testbeds; therefore, building a trust system, resource management, and virtualization of testbed services emerge as central themes. Firstly, the trust system provides authentication and authorization to enable users to access member WSN testbeds. KanseiGenie and SensLAB use login credentials on their portal accounts. WISEBED provides the SNAA (sensor network authentication and authorization) service based on a well-known distributed single sign-on system, Shibboleth. Once an account is provided, a user is able to authenticate himself by sending a web service request to the federator which represents a federated SNAA of several WSN testbeds and offers a secret authentication key in return. In IoT-LAB, all the objects including resources exist in a shared name space, organized in a hierarchy of authorities and subauthorities. The authentication mechanism is based on a public key infrastructure, where each object has a key pair and is associated with a signed certificate.

Secondly, resource management aims at a framework for transparent resource usages across the potentially heterogeneous resources in multiple member WSN testbeds. The framework shall allow users to be able to browse, discover, and reserve resources in a uniform way in spite of their heterogeneity. A minimum requirement is thus that the exposition of resource information to users and its reservation procedure should be general enough, while specialization is accommodated. KanseiGenie clearly addresses this requirement by using the ORCA control framework which is investigated intensively in the GENI project. ORCA uses a resource lease contract abstraction in which resource providers advertise or delegate their resources to broker intermediaries, and users request resources to those brokers. In WISEBED, resources are named using URN (uniform resource names) in which each node in the federation is uniquely identified by the combination of the member site ID, the sensor network ID, and the node ID. Resource specifications about node capabilities and attributes are posted in a user-readable text. A special client-side implementation, called a federator, federates multiple WSN testbeds, exposing their features as a single virtual testbed server instance. As a result, the resource reservation service provided by each individual WSN testbed is aggregated to form a virtual single service. SensLAB is based on a homemade portal, using LDAP to store its users and keys. Sharing resources among users is implemented using the OAR, which is a versatile resource manager for large clusters, particularly popular in the grid community. IoT-LAB takes different steps toward standard-based implementation. It adopts the slice-based federation architecture (SFA) which is a control plane architecture by the GENI initiative. Resource specifications in XML describe the set of resources made available by a WSN testbed. A registry manager manages authorities, users, slices, and resources. A registry is typically provided by an authority to manage all of these items. Aggregate manager APIs are provided to browse and reserve resources. An aggregate manager is typically deployed by an individual WSN testbed.

Finally, the virtualization aims at offering a uniform service abstraction across the member WSN testbeds. Generally, its implementation is based on web services. Deployed on their testbed servers, they are directly accessible by users. KanseiGenie provides an aggregate of aggregate manager
(AAM), the individual component managers (CM) for each device type, and an ORCA site authority module. Providing an iWSN interface as a Web services interface, WISEBED implements generic standard services and offers their APIs that range from managing federation, such as creating WISEBED-compliant custom WSN testbed, to conducting experiments by users. These APIs include SNA A APIs for authentication and authorization, RS APIs for reservation, WSN APIs for conducting, and finally controller APIs for collecting data from nodes. In SensLAB, web services, named experiment handler, serve as interaction points with the nodes in a site. With respect to each individual node, the experimentation handler provides firmware update, the energy consumption monitoring, and polling.

4.5. WSN Layout Domain

4.5.1. Native WSN Layout. Most of the WSN testbeds take this approach, as shown in Figure 6. In the native WSN layout, some WSN testbeds take a dedicated space, while others share laboratory rooms and corridors by installing nodes on their ceiling. This WSN layout inherently requires a large space; when preserving the normal radio transmission power in WSNs, distance between nodes in the layout ranges from 10 meters to 30 meters. Indeed, several large-scale WSN testbeds with over 100 nodes, for example, CPN [29], Indiya [33], and TWIST [24], are typically deployed over several floors. As a side effect, the cost of initial deployment and maintenance usually increases as the network size becomes larger. As a positive aspect, this approach provides a more realistic environment in terms of the space itself.

4.5.2. Grid WSN Layout. In this approach, nodes are placed on a relocatable square fixture with each side of approximately two to five meters. Compared with the native WSN layout, the grid WSN layout requires a quite smaller space, thereby being quite advantageous to scalability. Moreover, the installation and maintenance cost become relatively manageable. However, a difficulty with this approach is the need to control radio propagation distance such that an appropriate node density similar to the native WSN layout can be maintained. To this end, a naive RSSI control mechanism is used in Motelab [38] and Kansei [34]. As a more elaborated scheme, NetEye [39] instruments each node with a 3 dB SMA (Sub-Miniature version A) attenuator and an omnidirectional RH series monopole antenna. As another elaboration, ASSERT [28] implements a radio grid with both FPGA-based specialized hardware to control signal attenuation and coaxial cables to connect attenuators. It allows all the signal manipulations purely in the RF domain such that communication between nodes does not leak into the environment, thereby enabling experimenters to inject radio noises into the system and observe their influences on the system being studied.

5. Evaluation

In this section, we evaluate WSN testbeds listed in Table 1 with respect to the core requirements for WSN testbeds identified in Section 2. In Section 5.1, we describe the methodology used in our evaluation. In Sections 5.2, 5.3, and 5.4, we explain the set of criteria for each individual core requirement, which will be collectively used for assessing the fulfillment level of the core requirement. Influences exerted on each criterion by design approaches in our taxonomy are explained as well. In Section 5.5, we illustrate the influence assessment with respect to each criterion. Finally, the evaluation result of WSN testbeds obtained by applying the result of the influence assessment to each WSN testbed is explained in Section 5.6.

5.1. Evaluation Methodology. In the evaluation of WSN testbeds, we employ a methodology that consists of two steps: influence assessment and fulfillment assessment.

5.1.1. Influence Assessment. This is aimed at assessing inferences that each design approach in our taxonomy exerts on each criterion for the core requirements for WSN testbeds. At first, for each core requirement, that is, scalability, flexibility, and efficiency, we sort out a set of the criteria that will be collectively used to assess the fulfillment level of a WSN testbed for the core requirement. Each individual design approach in the taxonomy may have either a positive or a negative influence on a criterion. We thus examine and assess how each design approach exerts an influence on each individual criterion. Just for reference, the criteria and the result of the influence assessment are shown in Table 3. The procedure for the influence assessment is detailed shortly in Section 5.5.

5.1.2. Fulfillment Assessment. This is aimed at assessing how well a WSN testbed fulfills each of the core requirements. Given a WSN testbed, two kinds of information are used in the assessment of fulfillment levels. The first one is the classification result for the WSN testbed. In Section 3, we already classified the WSN testbed using our taxonomy. The classification result is shown, respectively, in Figure 6 and in Table 3. According to the classification result, the WSN testbed is featured with a set of four design approaches taken, respectively, from each individual design domain. The second kind of information is the result of the influence assessment which contains the influence type for each pair of a design approach and a criterion. The set of influence types obtained for the design approaches of the WSN testbed from the influence assessment result enables us to assess the fulfillment levels of the WSN testbed. This procedure is detailed shortly in Section 5.6.

5.2. Scalability Criteria. Scalability of WSN testbeds can be influenced by many factors from both technical and non-technical area. Among them, we select the three most critical ones, wiring, space, and cost, as the criteria for scalability.

5.2.1. Wiring. The wiring method for implementing a back-channel may restrict the number of nodes in the deployment. For example, as previously discussed, the coverage of active USB cables is within a maximum distance of 25 meters. An
active USB-hub infrastructure is limited to the maximum of 127 ports.

The wiring method for a WSN testbed is determined only by its back-channel organization; the other three design domains in the taxonomy do not affect the wiring method. Each design approach in the domain of the back-channel organization embodies its own inherent influential characteristics on the wiring criterion. Firstly, the native-BC approach exerts a negative influence on the wiring criterion since it uses either RS232c or USB in the back-channel. Secondly, the broadband-BC approach hardly exerts a negative influence on the criterion because Ethernet or WiFi, which is typically employed in this approach, is subject to neither distance nor port exhaustion. Finally, the hybrid-BC approach also hardly exerts a negative influence on the criterion. Typically, WSN testbeds with the hybrid-BC adopt USB as their native part and Ethernet or WiFi as their broadband part. Although a limitation is enforced by USB to the number of nodes per back-channel gateway, the limitation can be avoided by adding more back-channel gateways [24, 29, 33].

5.2.2. Space. Apparently, the space required for the deployment of a WSN testbed may hinder its deployment. Practically, it is difficult to afford a sufficient space due to either physical or budget limitation. Indeed, numerous WSN testbeds use corners or ceilings of their laboratories [36, 43]. To ensure a large-scale deployment, the space requirement should be as small as possible; the node density per unit space should be as high as possible. In this respect, the WSN layout is the only design domain that may influence the space criterion. Assessing the influence of each design approach in the WSN layout domain is very straightforward. The native WSN layout approach may exert a negative influence since it typically requires as much space as in the real-world deployment. On the contrary, the influence exerted by the grid WSN layout approach is negligible due to its far smaller space requirement.

5.2.3. Cost. Deployment cost serves potentially as a nontechnical barrier to test scalability. Of the four design domains, the back-channel organization can potentially exert influences on the deployment cost as it directly affects the cost per node. Of the four back-channel organizations, the broadband-BC with an external BGW controller can exert a substantially negative influence on the cost criterion mainly due to its need of a back-channel gateway per node. This is well addressed in the design of Indriya [33] which reports a comprehensive cost analysis between the broadband-BC and the hybrid-BC approach. It is very straightforward that the other approaches do not exert any substantially negative influence on the cost criterion.

5.3. Flexibility Criteria. Given a large body of WSN applications, flexibility of WSN testbeds can be influenced by many factors from both technical and nontechnical areas. Among them, we select the two most critical ones, variable node support and radio environment control, as the criteria for flexibility.

5.3.1. Variable Node Hardware. The variable node hardware is essential for flexibility. Three typical options exist for supporting the hardware variability. The first option is to use a heterogeneous mix of node platforms in the WSN field. For instance, TWIST [24] takes this option. The second option is to develop new nodes with hardware variability in mind. Several projects with stable funding sources take this option [28, 30, 43, 44]. Some of them take a modular design with a broad set of daughter modules [44], while the others integrate multiple functions in a single hardware [28, 30, 43]. Finally, the third option is to pursue federation [23, 25, 35, 40]. Federation is viewed as a macroscopic move to the hardware variability. For instance, WISEBED [25] introduces heterogeneity by using different classes of nodes from an embedded 8-bit microcontroller platform to a fully equipped PC.

In the design space, only a single design domain, the WSN field structure, may influence this criterion of the variable node hardware. Each design approach in this domain exerts its own inherent influence on the criterion. Firstly, a singular WSN field may exert a modestly negative influence on the criterion as it has to resort to the aforementioned first two options for hardware variability. These options do not guarantee a sufficient level of hardware variability. In case of the first option, under a heterogeneous mix of nodes, it is rather complex and even impractical to prepare and conduct experiments by remote users. The second option can provide the modest degree of hardware variability; however, it would never be sufficient in the presence of the large spectrum of WSN applications. Secondly, the clustered WSN field presents a more enlarged opportunity for hardware variability than the singular WSN field since it allows a set of heterogeneous WSN clusters. Finally, the federation is clearly the most viable option for hardware variability as a rich set of variable nodes can be provided from the member WSN testbeds.

5.3.2. Radio Environment. Capability to control the radio environment is essential for flexibility. It allows experiments to set up radio environment realistically for each unique target environment. Among others, radio inference and radio congestion are relatively more critical issues in the radio environment. Radio inference is usually controlled by noise injection, where external devices not participating in a WSN emit radio packet on the WSN radio band in order to tamper the WSN transmissions [34]. The control of radio congestion is typically based on the adjustment of radio transmission distance through transmission power control. Assuming the constant network workload per node, given a radio transmission distance, the radio congestion depends on the spatial density of nodes, whereas, given a constant spatial density, it depends on the radio transmission distance. As each node in a WSN testbed is rigidly placed in a fixed location, the control of the radio transmission distance provides a way to adjust radio congestion. In the architectural design space, only the WSN layout domain can potentially influence the radio environment control. In the WSN layout domain, the grid WSN layout approach is advantageous for radio environment control due to its compact deployment.
On the contrary, the extent of radio environment control is potentially restricted in the native WSN layout.

5.4. Efficiency Criteria. The efficiency of a WSN testbed can be greatly influenced by the experimental tools that the WSN testbed provides for experimenters. Over the experimentation lifecycle, a set of experimental tools is required to support experimental activities of experimenters. To name a few, it includes tools for reserving resources and parallel reprogramming in the experiment setup stage and tools for monitoring the progress of the experiment in the experiment execution stage. Across WSN testbeds, the provision of these tools varies depending on the investment to the implementation. However, there is a set of ingredients that render those tools versatile enough to fulfill the efficiency. Among them, we select the two most essential ingredients, resource transparency and real-time interactiveness, as the criteria for efficiency.

5.4.1. Resource Transparency. Resource transparency is to render an experimenter to have a local and private view for the resources of a shared remote WSN testbed as if the resources are privately owned by the experimenter. Resource transparency is required over the whole experimentation cycle. As an example, we examine resource reservation needed at the experiment setup stage. For a WSN testbed shared by many users, resource reservation becomes quite complex due to the need of a large volume of information on the WSN testbed resource. To name a few, the information includes the technical specification of each WSN node and both the physical and the radio layout of nodes. Therefore, a complete set of tools to browse, discover, request, and reserve resources is needed in order to render users to be able to transparently utilize WSN testbed resources. As an example, WeaverLab [44] employs a GUI representation where users can choose nodes by clicking corresponding icons on a graphical node layout. SensLAB [40] provides locale information in the form of \((x, y, z)\) coordinates for each WSN testbed node.

The resource transparency criterion is potentially influenced by the service architecture domain. It is partly because the improvement obtainable by implementation efforts is influenced by the architectural aspect of the service architecture. Ideally, a sufficient level of resource transparency is feasible only for private WSN testbeds based on the monolithic service architecture. For a shared WSN testbed based on either the tightly coupled or the loosely coupled architecture, significant efforts on both designs and implementations are entailed in order to provide users with an appropriate level of resource transparency.

5.4.2. Real-Time Interactiveness. Real-time interactiveness is another essential ingredient for efficiency. Over the experimentation lifecycle, experimenters can greatly benefit from real-time interactiveness. Firstly, interactive exposition of log data enable users to capture in a real-time manner the execution trace of interest. It is quite essential at the debugging stage of WSN protocols. Secondly, real-time experiment monitoring enable users to capture in a real-time fashion the knowledge of any permanent or temporary node breakdown caused by software crash or hardware malfunction. Such knowledge is especially valuable for supervising the progress of some critical tasks such as parallel reprogramming over the WSN nodes. It allows experimenters to have a confidence on the fidelity of the experiment result. Finally, real-time interactive exposition of log data facilitates the development of visualization tools to depict the data stream. Allowing users to capture the temporal variance of performance values in a real-time manner, these tools can greatly relieve users from the labor entailed when analyzing raw data.

The real-time interactiveness criterion is potentially influenced by the design domain of the service architecture. To provide real-time interactive interfaces for users, some architectural support is required. It is desired to extend flexibility with value-added functionalities such as GUI interfaces for real-time monitoring. Ideally, the ability for the client-side software to directly access the back-channel and get fed the log data stream is also needed. Monolithic and loosely coupled architectures are amenable to such architectural support. In tightly coupled architectures, the implementation of real-time interactiveness tends to be somewhat complex and costly. Unless the server-side software provides services to promptly push log data to the client-side software, real-time interactiveness may become an infeasible option. Customization or extension of value-added functionalities on the client-side software by a third party may not be allowed as well.

5.5. Influence Assessment. Based on discussions in the previous subsections, we assess inferences that each design approach in our taxonomy exerts on each criterion of the core requirements for WSN testbeds. Table 2 shows the result of the influence assessment.

Influences on a criterion by a design approach are classified into four different types. The first type indicates that the design domain, to which the design approach belongs, is not related to the criterion in terms of influence. This influence type is denoted by UNRELATED with symbolic representation “—” in the table. The second type represents a negative influence that prevents the design approach from fulfilling the criterion. This influence type is denoted by HARD-BARRIER with symbolic representation “X.” The third type represents another negative influence that becomes a barrier against the fulfillment of the criterion by the design approach, but the barrier can be practically resolved by appropriate measures. For example, as for a negative influence on the cost criterion, an increased expense can resolve the barrier. This type of influences is denoted by SOFT-BARRIER with symbolic representation “△.” Finally, the fourth type indicates any trivial or no influence that poses no barrier against the fulfillment of the criterion. This type is denoted by NO-BARRIER with symbolic representation “O.”

While discussions in the previous subsections make the assessment result in Table 2 self-explanatory, we point out several entries for elucidation. Firstly, while both the broadband-BC with an external BGW and the hybrid-BC
| Requirement | Criteria                  | Back-channel organization | Service architecture | WSN field structure | WSN layout |
|-------------|--------------------------|---------------------------|----------------------|---------------------|------------|
| Scalability |                         | Native | Broadband/on-board BGW | Hybrid | Monolithic | Tightly coupled | Loosely coupled | Singular | Clustered | Federated | Native | Grid |
| Wiring      | -                        | X     | O                    | O      | O          | —                 | —                 | —         | —         | —         | —       |
| Space       | -                        | —     | —                    | —      | —          | —                 | —                 | —         | —         | —         | —       |
| Cost        | —                        | O     | O                    | △      | O          | —                 | —                 | —         | —         | —         | —       |
| Flexibility | Variable H/W support     | —     | —                    | —      | —          | —                 | —                 | △         | O         | O         | —       |
|             | Radio environment control| —     | —                    | —      | —          | —                 | —                 | —         | —         | —         | —       |
| Efficiency  | Resource transparency    | —     | —                    | —      | —          | △                 | △                 | —         | —         | —         | —       |
|             | Real-time interactiveness | —     | —                    | —      | —          | △                 | △                 | △         | —         | —         | —       |

"—": unrelated, "X": hard-barrier, "△": soft-barrier, and "O": no-barrier.
# Table 3: Result of fulfillment assessment for WSN testbeds.

| WSN testbed | Native | Broadband/on-board BGW | Broadband/external BGW | Hybrid | Monolithic | Service architecture | WSN field structure | WSN layout | Scalability | Flexibility | Efficiency |
|-------------|--------|-------------------------|------------------------|--------|------------|--------------------|---------------------|------------|-------------|-------------|------------|
| ASSERT      | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 3           | 2           | 2          |
| CPN         | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 2           | 2           | 1          |
| EasyTest    | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 2           | 1           | 1          |
| Emulab      | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 2           | 1           | 1          |
| Indriya     | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 2           | 2           | 1          |
| Kansei      | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 3           | 2           | 1          |
| KanseiGenie | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 3           | 2           | 1          |
| KonTest     | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 1           | 3           | 3          |
| LabVIEW     | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 2           | 2           | 1          |
| Motelab     | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 2           | 2           | 1          |
| NetEyes     | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 3           | 2           | 1          |
| Sensei-uu   | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 2           | 1           | 2          |
| SensLab     | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 2           | 2           | 1          |
| IoTlab      | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 2           | 2           | 1          |
| Signelab    | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 0           | 1           | 3          |
| TWIST       | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 2           | 1           | 1          |
| WSNTB       | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 2           | 2           | 1          |
| WeaverLab   | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 3           | 2           | 2          |
| WISEBED     | o      | o                       | o                      | o      | o          | o                  | o                   | o          | 2           | 2           | 2          |
require external back-channel gateways, only the influence on the cost criterion by the former approach is assessed as SOFT-BARRIER due to its need for a back-channel gateway per node. Secondly, the influence on the resource transparency criterion is typed into SOFT-BARRIER for both the tightly coupled and the loosely coupled service architecture because they embody inherent complexities in supporting resource transparency. Thirdly, the influence exerted by the tightly coupled architecture on the real-time interactiveness criterion is assessed as SOFT-BARRIER. The reason for the assessment is the difficulty of implementing the real-time interactive monitoring capability due to both the server-based centralized data logging and the lack of direct interactions between client-side software and nodes.

The result of influence assessment in Table 2 enables us to get a clear perspective on the influential relation between design domains in the taxonomy and core requirements for WSN testbeds. On one hand, it reveals that each core requirement is influenced by a subset of design domains. Scalability is affected by both the back-channel organization and the WSN layout structure. Flexibility is affected by both the WSN structure and the WSN layout. Efficiency is influenced only by the service architecture. On the other hand, it shows that each design domain exerts influences mostly on a single core requirement. Only the WSN layout domain exerts influences on two core requirements, that is, scalability and efficiency.

5.6. Fulfillment Assessment of WSN Testbeds. Based on the result of influence assessment in Table 2 and the classification result for WSN testbeds in the left-hand columns of Table 3, we can evaluate how well each WSN testbed under examination fulfills each of the core requirements for WSN testbeds.

In the fulfillment assessment, we employ a 4-level grading system. The degree of fulfillment is graded into four levels from Level-0 to Level-3. A level value represents how much positively a WSN testbed fulfills a core requirement; 0 is the least and 3 is the best. More precisely, Level-0 indicates that a WSN testbed cannot fulfill a core requirement. On the contrary, Level-3 indicates that a WSN testbed can fulfill a core requirement without having negative influences on any criterion. Level-1 or Level-2 indicates that it can fulfill a core requirement, but modest investments have to be made to resolve the negative influences on a subset of criteria. Level-1 and Level-2 are distinguished by the number of the criteria with negative influences.

Given a WSN testbed, the fulfillment assessment of each core requirement is based on the following procedure. Firstly, if there is at least one criterion among the criteria associated with a core requirement on which the influence type is HARD-BARRIER, the fulfillment level for the core requirement is graded as Level-0. This is applied only to scalability since other core requirements, that is, flexibility and efficiency, do not have entries with HARD-BARRIER in Table 2. Secondly, unless a core requirement has any criterion assessed as HARD-BARRIER, its fulfillment is graded by the number of the criteria that have at least one entry marked as SOFT-BARRIER in the result of the influence assessment. If the number of such criteria is only one, the fulfillment level is assessed as Level-2. Otherwise, the fulfillment level is assessed as Level-1.

Table 3 shows the result of the fulfillment assessment. Overall, all the WSN testbeds under examination are not ideal without completely satisfying all the three core requirements. In case of scalability, three WSN testbeds have critical drawbacks with Level-0, while only two WSN testbeds get the highest fulfillment level. The rest of the WSN testbeds modestly fulfill the scalability with fulfillment Level-1 or Level-2. In contrast to the scalability, no single WSN testbed fails to fulfill either flexibility or efficiency; there is not a WSN testbed with Level-0, respectively, for each core requirement. This is an expectable result because both requirements are dependent more on either software or hardware implementation than on system architectures. Still, their fulfillment levels exhibit variance across the WSN testbeds, ranging from Level-1 to Level-3. As the assessment result of each WSN testbed is self-explanatory from the result of the influence assessment in Table 2, we do not expose discussions on each individual WSN testbed.

It is worth to note that the result of the fulfillment assessment for each WSN testbed in Table 2 reflects solely the architectural perspective of the WSN testbed subject to our taxonomy. Except Level-0, each value in the fulfillment assessment represents the lower bound in practice rather than the upper bound in practice. As noted earlier, a WSN testbed can completely overcome negative influences of the SOFT-BARRIER type which degrade the fulfillment level to either Level-1 or Level-2. Apparently, large-scale WSN testbeds built on a large volume of elaboration such as SensLAB [40] possess higher fulfillment levels in practice. Nevertheless, architectural merits and demerits of design approaches identified by our fulfillment assessment are clearly useful for systematic improvements of the current WSN testbeds.

From the architectural perspective, the evaluation result of current state-of-the-art WSN testbeds along with discussions in Sections 3 and 4 elucidates several important points. Firstly, efficiency of user experimentation is to be more valued in the WSN testbed design. This is a prerequisite to divert the current situation that fairly many well-known WSN testbeds are retired, stay dormant, or remain underutilized. Secondly, strategic consideration should be devoted to pragmatic aspects. Currently, there would be no serious technological barriers against securing the scalability of WSN testbeds. It is rather probable that nontechnological factors such as deployment cost and space become a hurdle against the scalability. In addition, we need to devote from the confusion that federation would contribute to the scalability. Different from general network disciplines such as the future Internet, federation in the context of WSN testbeds contributes primarily to hardware variability. The development of a full-blown federated WSN testbed is thus to be preceded by a comprehensive analysis on the trade-off between the utility value and the financial investment. Finally, more elaboration is required for some promising design approaches. To name a few, the grid WSN layout with delicate radio control is precious for scalability and flexibility but still remains in immaturity. The loosely coupled service architecture is
promising for efficiency but is not sufficiently explored. Design alternatives based on well-formed combinations of such design approaches would provide insight into solutions to current open problems such as narrowing the potential gap between experimental environments of WSN testbeds and real deployment environments to a reasonable level.

6. Conclusion

In a large-scale setting, appropriate WSN testbeds are essential for both researchers in academia and developers in industries for facilitating experiments of WSN protocols at all levels. Despite the decade-long WSN testbed research, active research and development are still ongoing particularly as a part of the current large-scale IoT research. This paper seeks to provide a framework to reason about the evolving large-scale WSN testbeds from a pragmatic and architectural perspective. A novel architectural taxonomy for general-purpose WSN testbeds proposed in this paper proves effective in examining and understanding architectural features of existing WSN testbeds. Moreover, it serves as an elegant means to qualitatively evaluate WSN testbeds for their responsiveness to a core set of requirements in the large-scale context. Architectural merits and demerits of design approaches identified by our evaluation for current WSN testbeds would clearly pave a way for systematic advances in current as well as future WSN testbeds. As future research, we plan to extend the current taxonomy by incorporating vertical specialized functionalities, such as mobility and health monitoring, and to elaborate the evaluation framework by applying a distinctive weight-based ranking to the set of evaluation criteria.

Conflicts of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the IT R&D program of MOTIE/KEIT. [10041145, Self-Organized Software platform (SoSp) for Welfare Devices].

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” Computer Networks, vol. 38, no. 4, pp. 393–422, 2002.
[2] I. Demirkol, C. Ersoy, and F. Alagöz, “MAC protocols for wireless sensor networks: a survey,” IEEE Communications Magazine, vol. 44, no. 4, pp. 115–121, 2006.
[3] Q. Wang, Y. Zhu, and L. Cheng, “Reprogramming wireless sensor networks: challenges and approaches,” IEEE Network Magazine, vol. 20, no. 3, pp. 48–55, 2006.
[4] N. A. Pantazis and D. D. Vergados, “A survey on power control issues in wireless sensor networks,” IEEE Communications Surveys & Tutorials, vol. 9, no. 4, pp. 86–107, 2007.
[5] R. Sugihara and R. K. Gupta, “Programming models for sensor networks: a survey,” ACM Transactions on Sensor Networks, vol. 4, no. 2, article 8, 2008.
[6] Y. Zhou, Y. Fang, and Y. Zhang, “Securing wireless sensor networks: a survey,” IEEE Communications Surveys & Tutorials, vol. 10, no. 3, pp. 6–28, 2008.
[7] W. Dong, C. Chen, X. Liu, and J. Bu, “Providing OS support for wireless sensor networks: challenges and approaches,” IEEE Communications Surveys & Tutorials, vol. 12, no. 4, pp. 519–530, 2010.
[8] M. Garcia, D. Bri, S. Sendra, and J. Llret, “Practical deployment of wireless sensor networks: a survey,” International Journal on Advances in Networks and Services, vol. 3, no. 1-2, pp. 170–185, 2010.
[9] B. Musznicki and P. Zwierzynkowski, “Survey of simulators for wireless sensor networks,” International Journal of Grid and Distributed Computing, vol. 5, no. 3, pp. 23–50, 2012.
[10] U. Colesanti, C. Crociani, and A. Vitaletti, “On the accuracy of OMNeT++ in the wireless sensornet works domain: simulation vs. testbed,” in Proceedings of the 4th ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, pp. 25–31, 2007.
[11] K. Pawlikowski, H.-D. J. Jeong, and J.-S. R. Lee, “On credibility of simulation studies of telecommunication networks,” IEEE Communications Magazine, vol. 40, no. 1, pp. 132–139, 2002.
[12] M. O. Farooq and T. Kunz, “Wireless sensor networks testbeds and state-of-the-art multimedia sensor nodes,” Applied Mathematics and Information Sciences, vol. 8, no. 3, pp. 935–940, 2014.
[13] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton, and T. Razafindralambo, “A survey on facilities for experimental internet of things research,” IEEE Communications Magazine, vol. 49, no. 11, pp. 58–67, 2011.
[14] L. Steyn and G. Hancke, “A survey of wireless sensor network testbeds,” in Proceedings of the IEEE AFRICON, pp. 1–6, September 2011.
[15] A. Dwivedi and O. Vyas, “An exploratory study of experimental tools for wireless sensor networks,” Wireless Sensor Network, vol. 3, no. 7, pp. 215–240, 2011.
[16] G. Wagenknecht, M. Anwander, T. Braun et al., “TARWIS: a management architecture for heterogeneous wireless sensor networks,” in Proceedings of the 6th International Conference on Wired/Wireless Internet Communications, pp. 177–188, 2008.
[17] S. Bapat, W. Leal, T. Kwon, P. Wei, and A. Arora, “Chowkidar: a health monitor for wireless sensor Network testbeds,” in Proceedings of the 3rd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom’07), pp. 1–10, Lake Buena Vista, FL, USA, May 2007.
[18] J. Kim, J. Lee, and J. Yun, “M2M service platforms: survey, issues, and enabling technologies,” IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 61–76, 2014.
[19] C. Siaterlis, B. Genge, and M. Hohenadel, “EPIC: a testbed for scientifically rigorous cyber-physical security experimentation,” IEEE Transactions on Emerging Topics in Computing, vol. 1, no. 2, pp. 319–330, 2013.
[20] J. Bernat, “SmartSanter : the path towards the smart city vision,” in Proceedings of the 1st ETSI M2M European Telecommunications Standards Institute Machine to Machine Workshop, Sophia Antipolis, France, October 2010.
[21] J. Polastre, R. Szewczyk, and D. Culler, “Telos: enabling ultra-low power wireless research,” in Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN ’05), pp. 364–369, April 2005.

[22] Tmote Sky Data Sheet, pp. 366–379, Moteiv, San Francisco, Calif, USA, 2006.

[23] O. Rensfelt, F. Hermans, P. Gunningberg, and L.-A. Larzon, “Repeatable experiments with mobile nodes in a relocatable WSN testbed,” in Proceedings of the 6th IEEE International Conference on Distributed Computing in Sensor Systems Workshops (DCOSSW ’10), pp. 1–6, Santa Barbara, Calif, USA, June 2010.

[24] V. Handziski, A. Köpke, A. Willig, and A. Wolisz, “TWIST: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks,” in Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing, pp. 63–70, May 2006.

[25] I. Chatzigiannakis, S. Fischer, C. Koninis, G. Mylonas, and D. Pfisterer, “WISEBED: An open large-scale wireless sensor network testbed,” in Sensor Applications, Experimentation, and Logistics, pp. 68–87, Springer, Heidelberg, Germany, 2010.

[26] C. Elliott and A. Falk, “An update on the geni project,” ACM SIGCOMM Computer Communication Review, vol. 39, no. 3, pp. 28–34, 2009.

[27] A. Gavras, A. Karila, S. Fdida et al., “Future internet research and experimentation: the FIRE initiative,” ACM SIGCOMM Computer Communication Review, vol. 37, no. 3, pp. 89–92, 2007.

[28] E. Nourbakhsh, J. Dix, P. Johnson et al., “ASSERT: a wireless networking testbed,” in Proceedings of 7th International Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities, pp. 209–218, 2011.

[29] J. Kowalczuk, M. C. Vuran, and L. C. Pérez, “A dual-network testbed for wireless sensor applications,” in Proceedings of the 54th Annual IEEE Global Telecommunications Conference: “Energizing Global Communications” (GLOBECOM ’11), pp. 1–5, December 2011.

[30] Z. Zhao, G.-H. Yang, Q. Liu, V. O. K. Li, and L. Cui, “EasiTest: a multi-radio testbed for heterogeneous wireless sensor networks,” in Proceedings of the IET International Conference on Wireless Sensor Network (IET-WSN ’10), pp. 104–108, November 2010.

[31] D. Johnson, D. Flickinger, T. Stack et al., “Emulab’s wireless sensor net testbed: true mobility, location precision, and remote access,” in Proceedings of 3rd International Conference on Embedded Networked Sensor Systems, p. 306, 2005.

[32] D. Johnson, T. Stack, R. Fish et al., “Mobile emulab: a robotic wireless and sensor network testbed,” in Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM ’06), pp. 1–12, April 2006.

[33] M. Doddavenkatappa, M. Chan, and A. Ananda, “Indriya: a low-cost, 3D wireless sensor network testbed,” in Proceedings of 7th International Conference on Testbeds and Research Infrastructure and Development of Networks and Communities, pp. 302–316, 2011.

[34] E. Ertin, A. Arora, R. Ramnath et al., “Kansei: a testbed for sensing at scale,” in Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing, pp. 399–406, 2006.

[35] M. Sridharan, W. Zeng, W. Leal et al., “KanseiGenie: software infrastructure for resource management and programmability of wireless sensor network fabrics,” in Next Generation Internet Architectures and Protocols, pp. 275–300, Cambridge University Press, 2010.
