HAL
archives-ouvertes.fr

A Survey on (mobile) wireless sensor network experimentation testbeds

Anne-Sophie Tonneau, Nathalie Mitton, Julien Vandaele

▶ **To cite this version:**

Anne-Sophie Tonneau, Nathalie Mitton, Julien Vandaele. A Survey on (mobile) wireless sensor network experimentation testbeds. DCOSS - IEEE International Conference on Distributed Computing in Sensor Systems, May 2014, Marina Del Rey, California, United States. 2014. <hal-00988776>

HAL Id: hal-00988776

https://hal.inria.fr/hal-00988776

Submitted on 18 Jul 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Survey on (mobile) wireless sensor network experimentation testbeds

Anne-Sophie Tonneau, Nathalie Mitton, Julien Vandaele
Inria Lille - Nord Europe - {firstname.lastname}@inria.fr

***Abstract*—With the development of new technologies, these last years have witnessed the emergence of a new paradigm: the Internet of Things (IoT) and of the physical world. We are now able to communicate and interact with our surrounding environment through the use of multiple tiny sensors, RFID technologies or small wireless robots. This allows a set of new applications and usages to be envisioned ranging from logistic and traceability purposes to emergency and rescue operations going through the monitoring of volcanos or forest fires. However, all this comes with several technical and scientific issues like how to ensure the reliability of wireless communications in disturbed environments, how to manage efficiently the low resources (energy, memory, etc) or how to set a safe and sustainable maintenance. All these issues are addressed by researchers all around the world but solutions designed for IoT need to face real experimentations to be validated. To ease such experimentations for IoT, several experimental testbeds have been deployed offering diverse and heterogeneous services and tools. This article studies the different requirements and features such facilities should offer and survey the different experimental facilities currently available for the community, the different hardware used (as sensors and robots) and the scope of their services. We expect this survey assist a potential user to easily choose the one to use regarding his own needs. Finally, we identify existing gaps and difficulties and investigate new directions for such facilities.**

I. Introduction

The Internet of Things (IoT) results from the combination of technological progresses and the new habits and needs humans have developed facing it. We are now able to communicate and interact with our surrounding environment through the use of multiple tiny sensors, RFID technologies or small wireless robots. This allows a set of new applications and usages to be envisioned ranging from logistic and traceability purposes to emergency and rescue operations going through the monitoring of volcanos or forest fires. Nowadays, technologies have improved, becoming more complex and efficient, and new technological challenges have emerged. The applications developed on top of these technologies need to be tested and improved before being exposed to the reality. Efficient simulation tools are useful to help in the design of IoT applications, since they offer a quick and flexible way to test the behaviour of an application in a repeatable manner. But simulation leads to assumptions on several parameters of the environment, that is a cause of uncertainty. IoT applications are seriously influenced by unpredictable events and physical characteristics, very difficult to simulate. There is a strong need to deploy applications in a real-life like context, therefore conducting experiments on real hardware, at large-scale, and to benefit from appropriate tools for experimentation management. But experimenting on large scale is a fastidious, expensive and time-consuming task. Therefore, several testbeds have been deployed all around the world to allow faster experimentations, with various sizes, hardware, topologies, and degrees of flexibility. Some facilities focus on large-scale deployment, others on mobility. Some are quite specialised, others more flexible, allowing experimentation of purely technical issues as well as higher level applications.

This survey first describes the requirements a facility should take into account and the challenges faced up for such deployments and reviews the existing available IoT testbeds with regards to these requirements. Of course, we do not pretend to be exhaustive since the number of testbed initiatives in the world is huge. We only focus on the most meaningful and active testbeds. Likewise, some additional functionalities of the mentioned testbeds may have been omitted for the sake of clarity and coherency. The interested reader is invited to refer to the cited papers for further information. We expect this paper constitutes a tool to assist an experimenter to find the adequate facility that better matches his specific needs.

The remainder of the paper is as follows. Section II describes and sets the different requirements for an IoT experimental testbed. Section III surveys the existing facilities and discusses their main purposes and functionalities. Finally, Section IV provides some concluding remarks.

II. Requirements and challenges

Because of the diversity of wireless networking issues and applications, a wireless sensor network (WSN) testbed is expected to be flexible, to support various network topologies and network layer protocol options in order to allow the greatest number of applications. The facility must enable the design of as much realistic IoT experimentations as possible, in terms of scale, behaviour, functionalities, environment and constraints. We have gathered these features into the following categories : *(1)* Experimentation, *(2)* Hardware features, *(3)* Mobility and *(4)* Extra features, that we detail in the following.

A. Experimentation

From the user perspective, the services and tools offered to design and interact with the experiment should be easy and intuitive to take in hand.

Specification: The first step of an experimentation is the specification, *e.g.* the selection of the adequate resources in terms of number, type or other properties, but also specifying the programs to upload, and the data to be collected. The way to set up and validate a configuration is an important feature.

Interaction: The testbed should provide proper interfaces to interact with the devices, and with the ongoing experiment in order to follow its progress, adjust parameters or debug. While most of the existing testbeds are accessible via web interfaces, only a few provide ssh front-end to access resources or both. Some expose their resources and functionalities as web services, offering the user the possibility to develop their own

client applications on top of the web services API. Nevertheless security and availability questions have to be taken into account. During an experiment, it is also necessary to facilitate the access to sensors and to network-related metrics, such as the delay, throughput, overhead or energy consumption. Finally, testbeds should provide some visualisation tools for experimental data collection and data analysis.

Repeatability: There is a need to be able to repeat experiments within and across different testbeds. For instance, to analyse the influence of one specific parameter, several experiments should be run with this parameter varying. At the experiment level, repeatability can be achieved by standardising the experiment specification and recording it, as well as the firmwares to upload on the nodes. Even if the total real-world conditions replication is not possible (because of external interference and human activity), this repeatability requirement can be partially overcome by keeping the experimenter updated about the environmental conditions, and collecting traces, to help him/her to contextualise his/her results.

Simulation: When designing a wireless sensor network application, it is obvious that emulation and simulation are essential steps to put to the test the viability of a solution. Some efficient simulation and emulation tools exist and are widely used, like WSNet[1] or NS-2/3[2]. However, they suffer from a lack of accuracy in capturing realistic environmental conditions, like radio propagation. Some testbeds include simulation tools to alleviate the design of experiments, and to verify the consistency of a protocol or algorithm, before putting it into practice by using the testbed hardware. However, an interesting approach, detailed later in Section III-A, is to combine simulation, emulation and physical elements together into a single testbed, in order to gain flexibility on the scale and the offered configurations and to lower the trade-off between repeatability, reliability and scalability.

B. Hardware features

The hardware constitutes one of the main concerns of the user, since the goal of an experimental platform is to facilitate access and testing on real equipment, under realistic conditions and constraints. The hardware used have to match as much as possible the needs of targeted applications.

Heterogeneity: The IoT concept relies on the key feature of heterogeneity, meaning that devices are based on different technologies with various capabilities, with various sensor types. This required diversity in devices, coming with diverse drivers, toolchains or operating systems, drives to the need of easy programming and configuration of these heterogeneous devices with means for programming the devices, like drivers, communication libraries, or operating systems porting.

Scale: Another important property of an IoT testbed is the scale, *i.e.* the number of devices available for experimentation. IoT systems and technologies apply on much more than a few dozen of devices. A facility should be scalable, with easy extension and update of the hardware, and the possibility to include more recent devices, in a plug-and-play manner.

Federation: Some testbeds are multi-site, *i.e.* the resources of the testbed are distributed over different locations, with a unique access point to all the nodes. Some others are federated with other IoT testbeds. This enables the addition of capabilities for experimentation and the access to more heterogeneous devices, with scaling up. The federation of existing testbeds requires a common framework, a layer built on top of all existing infrastructures, allowing the user to authenticate and reserve resources on every interconnected testbed simultaneously. This survey looks over main existing both multi-site and federated testbeds.

C. Maintenance

The testbed maintenance occurs at several frequencies. A daily maintenance needs to be considered to ensure the good functioning of the testbed and experimentation. A more general maintenance is also needed to verify that the hardware is operational and that the software architecture still offers appropriated services. Finally, to amortise the platform deployment, some hardware makeover should be carefully scheduled to allow the longest lifetime to the facility. To alleviate the maintenance, the testbed design needs to lower human intervention as much as possible, in order to limit the resulting cost of extra human resources involved to prevent hardware malfunctioning or even damages and take care of the health of the testbed. In parallel, a constant monitoring of the software is important to prevent from crash and constantly update firmwares and services.

D. Mobility

IoT applications involve mobile devices that collect information on the environment, or cooperate with each other, which leads to the design and implementation of robotic and automation systems.

Undergone vs Controlled Mobility: Several types of mobility exist: undergone or controlled mobility. We refer to undergone mobility for a device which is embedded on an object/person that cannot be controlled by the device itself. Such undergone mobility can either be non predictive, in the case of human or animal carriers, or predictive, for example if the devices are carried by public transport, like a bus, which has a predefined and known journey. While mobility offers larger possibilities in terms of applications, handling such a feature and providing adequate support for controlling an experiment and exploiting results is a real challenge. Some testbeds have introduced undergone mobility, predictive or not [1], [2]. Some others provide controlled mobility [3], which implies additional constraints for locating and charging.

Autonomous charging & localisation: Because of the mobility, potential collisions may happen, causing damages to the hardware and interrupting a running experiment prematurely. Therefore, the testbed should autonomously run, with a remote access provided to users that allow them to perform experimentation without compromising the safety of the hardware thus the continuity of the experiment. This leads to several material constraints: the robot should be self-rechargeable (and empowers the embedded device at the same time) and able to locate and reach the charging system. Thus, mobility requires autonomous localisation and path planning with obstacles avoidance. Therefore, an accurate positioning mechanism is needed to overcome these issues. Section III-C describes some existing localisation solutions.

[1] http://wsnet.gforge.inria.fr
[2] http://www.nsnam.org

Software management & tools: An essential requirement for IoT testbeds is to ease the implementation of scenarios, and the interactions with the hardware. For robot control, there are several drivers and frameworks that provide services, hardware abstraction, low-level device control, and so on. Some testbeds use these control interfaces, some others have developed their own middleware. Testbeds should provide various mobility models that are ready-made and possibly customizable. Finally, it is paramount, when conducting experiments including mobile devices, to dispose of visualisation tools that display the running experiment state, but also past experiment configurations, like the paths the mobile devices had actually followed, and to ensure repeatability.

III. Survey of existing testbeds

In this section, we discuss how today's testbeds answer the requirements and address the challenges detailed above. We will compare them in terms of capabilities and features they offer, in accordance with the categories of needs previously described. We browse wireless sensor testbeds technologically well-advanced or widely used, with a focus on mobility. Table I sums up the hardware and key features of the studied testbeds.

A. Experimentation

Specification: Getting familiar with a new facility comes with the specification of the experiment. Most of the existing testbeds (FIT IoT-LAB, TWIST, Kansei, NetEye, SmartSantander or WISEBED) provide web interfaces for job scheduling, to specify the resources needed, set up and program the nodes to define their behaviour and the data to be collected. FIT IoT-LAB (IoT-LAB in short in the following) offers two ways to select nodes: either by Id, or by properties (site, radio chip, mobility, etc). FlockLab, ISRobotNet, or IoT-LAB use XML or JSON files to store the configuration of the nodes and the specification of an experiment. WISEBED has built its own generic XML-based language called WiseML, for experiment and testbed description, configuration, and results storage while others use databases. To handle the concurrent reservations, NetEye, IoT-LAB and WISEBED use a first-come-first-serve approach, meaning that the first user submitting an experiment on available resources gets the access.

Interaction: For interactions with an experiment, several different approaches exist. Every testbed provides at least a command-line tool, and more commonly a web-based front-end (W-iLAB.t). The very popular MoteLab, built at Harvard University, has been decommissioned, but its service framework remains a basis for various other testbeds like *e.g.* INDRIYA. Using the web interface that INDRIYA has inherited from MoteLab, users can upload, monitor, and control their jobs remotely and in real-time. Regarding the testbeds focused on mobility, the need of an efficient and interactive visualisation tool is crucial. With this aim, CONET-IT has developed an Integrated Testbed GUI that allows the visualisation of the experiment and data log (programming the WSN nodes and robots, graphically setting waypoints to the robots, accessing the camera and laser views, data logging, etc). Mint-m's network/experiment management subsystem, called MOVIE (Mint-m cOntrol and Visualisation InterfacE) affords a user full interactive control over the testbed as well as real-time visualisation of the testbed activities. In addition to a web-based front-end, for more advanced and specific interactions with the testbed, some expose their services via a web services API, either RESTful (WISEBED, SmartSantander and IoT-LAB) or SOAP-based (WISEBED, SmartSantander). Users are then able to develop their own tools on top. WISEBED provides a selection of open-source Web- and Desktop Clients, that can be adapted by users for their specific needs.

In order to program the nodes before or during an experiment, a user may need to log on the testbed's server. To this purpose, TWIST or IoT-LAB provide ssh access to start, stop, reset, update the nodes, and read or write on the serial links. CONET-IT offers a virtual private network to interact with the Integrated Testbed (IT) during the experiment. SmartSantander and WISEBED have developed a set of command-line scripts to control and interact with the experiment, automate, repeat experiments, and even programmatically analyse, convert and process output from the nodes. In SmartSantander, least energy constraint nodes can be reprogrammed for the need of the experiment while some "service nodes" only produce data on top of which users can develop new services. MOTEL uses XML files loading with predefined commands to be sent out at some times to the network, as well as a graphical tool to send commands to the nodes wirelessly and to get status information feedbacks and log files.

Finally, data collection and experiment analysis are essential steps of the experimentation process. IoT-LAB gathers resulting periodic measurements coming from sensors into CSV files stored into the user's home directory. These files are then easily parsable and analysable using graphical tools like Plot. SmartSantander platform stores in its database all the observations and measurements sent by the different IoT nodes, containing live and historic information. Furthermore, it provides a tool called TMON, a Java based experimentation environment, allowing, amongst other things, the visualisation of traces and live results. As for INDRIYA and W-iLAB.t, all messages and other data are logged to a database which is presented to the user upon job completion and then can be used for processing and visualisation.

Repeatability: For repeatability, most testbeds save the experiment description, and propose, like IoT-LAB, to reload the configuration. SmartSantander and WISEBED made the automated repetition possible thanks to experimentation scripts.

Simulation: CONET-IT makes mandatory to go through simulation on Gazebo before using the facility. INDRIYA just advises users to prototype their application using TOSSIM and then upload it through the web interface. A more advanced functionality is provided by TWIST: the Cooja-TWIST plugin lets experimenters use the testbed directly from Cooja.

Another approach is the emulation of events, characteristics, into the experiment in order to extend hardware capabilities, when some requirements are not physically supported on the actual nodes. Examples of emulated events (TWIST, SmartSantander and FlockLab) are the node death, new nodes, allowing the modification of the network topology. Kansei and IoT-LAB make available the event injection, WISEBED goes further, providing the functionality to create virtual links between nodes, thus dynamical experiment specification, composed of physical or virtual nodes and virtual links. These hybrid testbeds can be accessed and controlled by common tools, in a transparent way.

Testbed	Hardware Summary	Notes
IoT-LAB [2]	2728 heterogeneous, specifically developed motes located in 6 different sites	SensLAB follow-up, repeatable mobility via electric toy trains, energy consumption measurement, multi-site experiments, part of FIT
TWIST [4]	204 motes (102 TmoteSky + 102 eyesIFX) spread across 3 floors	Supports flat and hierarchical setups, node death and addition emulation, cost-effective and open solution which can be reproduced by others
Kansei [5]	210 XSM motes: large grid-like structure of motes evenly distributed	Supports various platforms such as Extreme Scale Motes (XSMs), TelosB, Imote2 and Stargates, event injection at GW and mote level
NetEye [6]	130 TelosB motes, indoor	Static 3db attenuators are attached to the mote antennas for multi-hop network and different power levels
SmartSantander [1]	20000 (fixed, mobile & smartphone) sensors over 4 countries, indoor and outdoor	Most advanced testbed in terms of hardware, scale, functionalities. Mobility via public buses. Multi-site and real-life experiments
WISEBED [7]	750 motes, mainly iSense, MicaZ, and Pacemate, SunSPOT, and TelosB motes	Large federation that includes some SmartSantander testbeds, simulator engines that create virtual testbeds
DES-Testbed [8]	95 nodes: embedded PC board with up to 3 IEEE 802.11 cards, and wireless sensor	Virtualizer running several virtual machines that recreate the testbed topology and its lossy links. 1 mobile DES-Node node on a Roomba
FlockLab [9]	30 observers and 1 server spread across one level of a building at ETH Zurich	Sensor node pairing with dedicated hardware for monitoring and simulation
INDRIYA [10]	139 TelosB motes spread across 3 floors	Experiment prototyping with TOSSIM simulation environment, web-based interface designed based on Harvard's MoteLab interface, nodes replacement with Arduino motes
w-iLab.t [11]	200 Tmote Sky + 60 more powerful nodes in 2 different locations	Different types of wireless nodes: sensor nodes, Wi-Fi based nodes, sensing platforms, and cognitive radio platforms, uses the Emulab software at is base

Testbed	Sensors	Robot platforms	Localisation solution	Software	Remote access	Notes
CONET-IT [3]	WSN of 21 static nodes: TelosB, MicaZ, Mica2, Iris	5 Pioneer 3-AT + Aspire One ZG5 netbook + Hokuyo 2D Laser + Kinect + Wireless a/b/g/n Bridge	Vision-based with AMCL, decentralised	Player Stage (ROS porting planned)	yes, single user	random motion with obstacle avoidance
RoombaNet [12]	WSN of 6 static nodes	6 Roomba + mobile controller, extension up to 14 robots planned	Odometry and orientation sensor-based positioning	Wiselib ported to the sensor node platform	No, open to scheduled remote users	2 mobility models: random & semi-random Wiselib extension to support Roomba control software planned
MOTEL [13]	TelosB, MicaZ, Scatterweb	22 e-puck + Thymio II	Vision-based with cameras, centralised	FLEXOR MuRobA	No	No communication sensor-robot, software for interactive control, not permanently installed
Mint-M [14]	Wireless network node supporting 802.11 interfaces	12 Roomba	Vision-based with cameras, centralised	MOVIE (Mint-m cOntrol and Visualization InterfacE)	No	Testbed can operate without human intervention for weeks in a 24x7 fashion, deployable in a limited physical space (radio signal attenuation), strong simulation tool
ISRobotNet [15]	?	4 Pioneer AT + 1 ATRV-Jr	Vision-based with cameras, centralised	YARP networking software	No	Fully decentralized use of the resources

TABLE I. SURVEYED WSN TESTBEDS (TOP) AND TESTBEDS FOCUSED ON MOBILITY (BOTTOM)

B. Hardware features

Heterogeneity: Facilities have deployed a huge variety of hardware, where motes can be off-the-shelf as well as custom-built for a specific need. NetEye and DES-Testbed offers a single kind of device. NetEye has deployed TelosB motes, with an MSP430 microcontroller, a CC2420 radio and usual light, temperature and humidity sensors. DES-Testbed offers custom MSB-A2 sensor nodes (developed at Freie Universitat Berlin), equipped with a Chipcon CC1100 transceiver as well as temperature and humidity sensors. Most of the testbeds offer heterogeneous types of devices: TWIST has deployed TmoteSky and EyesIFX nodes, both integrating a MSP430 microcontroller. W-iLAB.t also offers TmoteSky, together with other more powerful motes. Furthermore, Kansei infrastructure includes 802.11, 802.15.4, and 900 MHz Chipcon CC1000 radios, as well as diverse sensor nodes, including XSM (based on Mica2 platform), TelosB, Imote2 and Stargates. INDRIYA has recently replaced a few dozen of its TelosB motes with Arduino motes to follow this heterogeneity requirement. In IoT-LAB, custom motes are deployed, mostly based on TI MSP430, but also ARM Cortex M3 and ARM Cortex A8, more powerful motes and equipped with other sensors such as accelerometer, magnetometer and gyrometer. SmartSantander and WISEBED offer far more heterogeneous devices, since deployment is not only indoor, as for all previously described testbeds, but also outdoor and in-vivo. SmartSantander offers smart city services, involving the citizens into the experimentation loop. With this aim, IEEE 802.15.4 devices, GPRS modules, and joint RFID tag/QR code labels are deployed both at static locations (streetlights, façades, bus stops) as well as on-board on mobile vehicles (buses, taxis). It also provides indoor deployment with iSense, TelosB, and Pacemate motes, as well as Sunspots. WISEBED provides as many heterogeneous devices, and a wide range of sensor types, ranging from most commonly used temperature sensors to more sophisticated Anisotropic Magnetoresistance (AMR) sensors. TWIST, NetEye and INDRIYA impose TinyOS while IoT-LAB, SmartSantander and WISEBED have no mandatory operating system. In addition, IoT-LAB provides drivers, MAC layers, communication libraries, and OS porting for Contiki, FreeRTOS, TinyOS, and RIOT.

Scale: We observe that most of the single location testbeds (*e.g.* INDRIYA, TWIST) feature a limited number of nodes (up to 200 nodes), most likely due to cost and space constraints. However, Kansei succeeded to deploy about 700 motes in a single location, an indoor grid-like structure of motes evenly distributed on tables within a warehouse. The largest testbeds are the distributed ones, such as WISEBED which count up to 750 motes, while IoT-LAB and SmartSantander have deployed several thousands of nodes.

Federation: In most cases, large-scale feature is made possible by the deployment of multi-site testbeds, or by the federation of existing testbeds. IoT-LAB testbed is multi-site, spread across 6 different locations in France and gives forward access to 2728 wireless sensor nodes. It is composed of previously named SensLAB testbeds [2] plus latest deployments of relatively new wireless sensor devices. One master server manages several clients (each testbed site) and users access the testbed through a REST API exposed by the master. It is an actual distributed testbed which offers the functionality to experiment on several different locations on different hardware at the same time. WISEBED federation comprises about 750 devices supporting a range of sensor modalities. On each of the 9 sites, a server exposes the testbed capabilities to the outside through an iWSN interface, providing a uniform access. Finally, the largest federation in terms of number of devices is SmartSantander. Indoor, outdoor as well as in-vivo devices are deployed across 4 different cities in Europa: Belgrade (Serbia), Guildford (UK), Lübeck (Germany) and

Santander (Spain). Some of the testbeds are accessible through the already mentioned WISEBED experimental facility feature thus are also part of this federation.

C. Mobility

In this section, we examine the key features of mobility provided by different testbeds. The majority of the facilities does not offer any type of mobility, some others have introduced mobile nodes into an existing grid of fixed nodes, or are currently dealing with that issue.

Undergone vs Controlled Mobility: Undergone mobility, as described in Section II-D, is provided by some testbeds as in SmartSantander where nodes are embedded into public transportation vehicles. These nodes are remotely accessible for experimentation and can be used to determine the location of the vehicles, estimate arrival time to bus stops, or make atmospheric measurements. In IoT-LAB, some nodes are embedded on 4 electrical toy trains manoeuvring on 2 separated circuits and on more than 200 robots (wifibots and Roomba). In these specific cases, the user can choose the speed of the trains, and/or the mobility pattern of each robot.

Offering controlled mobility is an even more difficult issue. Some testbeds have introduced controlled mobility by means of robot platforms carrying the nodes, but do not offer remote access to this infrastructure except IoT-LAB and CONET-IT. To date, most of testbeds have not deployed more than a few robots. While enlarging the scale is a goal for many of them, they usually adopt an incremental approach, solving technical problems and validating it. For instance, on the DES-Testbed, only one mobile node embedded on a Roomba is running, still as a prototype, but it is planned to extend it with other robots. Roomba is the most widely used robot for introducing mobility in testbeds, since it is low-cost and self-operating, and has auto-recharging capabilities. WISEBED, IoT-LAB, MINT-m and RoombaNet have a few of them available either occasionally (WISEBED) or permanently (RoombaNet plans to extend its fleet to 12 robots). But for all these testbeds, scheduling and timeslices to the remote users remain to be clarified. ISRobotNet uses 5 Pioneer 3-AT and ATRV-Jr robots, while MOTEL has 22 e-puck and Thymio II robots running. A more advanced testbed, remotely accessible and autonomous is CONET-IT, which has deployed about a hundred heterogeneous wireless sensor motes, some of them piggybacked on 5 Pioneer 3-AT mobile robots. Some extension in terms of numbers of sensors and robots are scheduled. However, one experimenter must book the entire testbed. Going further, IoT-LAB also offers several nodes embedded on robots (up to 200 robots), either Roomba, Turtlebot2 or Wifibot. These robots and their embedded nodes are included in the reservation and scheduling system and are managed as any other resource. Robots can be used either to provide undergone mobility to their embedded nodes or as are, giving full control of them to the user, also being able to interact with their embedded node.

Autonomous charging & localisation: Testbed providers that use Roomba robots have modified their auto-recharging circuitry to power up both the Roomba and the carried wireless sensor node. MINT-m has even developed a residual power algorithm that is able to estimate the amount of energy left, and a recharge scheduling algorithm, thus allowing no human intervention for weeks. The Turtlebot2 deployed in IoT-LAB are off-the-shelf robots targeted for research and education purposes, and provide functionalities like auto-recharging feature. In contrast, a specific auto-recharging solution had been designed by IoT-LAB providers for the Wifibot robots. As for other testbeds, human intervention is almost always required at one point or another to recharge the robots.

An accurate positioning mechanism is necessary for the robots to move autonomously in an area and to avoid obstacles. Several approaches exist, either centralised or distributed. RoombaNet controls the directions and distances using odometry and additional orientation sensors. MINT-m, MOTEL and ISRobotNet rely on a vision-based system with overlooking cameras mounted on the ceiling. A central server computes images and identifies the positioning of each robot. In MINT-m, the tracking server is a cluster of PCs that periodically receives snapshots of the entire testbed as captured by the 6 cameras. On MOTEL, the system tracks the coloured markers fixed on top of the robots. CONET-IT has opted for a distributed approach for localisation: a robot is equipped with a laser, and uses the probabilistic system AMCL (Adaptive Monte-Carlo Localisation) to localise itself. The algorithm uses a particle filter captured by the laser as well as the odometry data to track the pose of the robot into the map of the environment.

Software management & tools: On the software side, many testbeds have developed their own tools. RoombaNet has ported Wiselib on its sensor nodes and developed a driver that encapsulates Roomba's basics movements, allowing the robot to move on a given distance or angle. This will enable users to control the movements of the robots. MOTEL has developed FLEXOR that run on top of TinyOS. FLEXOR is platform-independent and has extensive graphical support to program and manage WSNs. The control of the robots is handled by a multi-robot architecture for coordinated mobility called MuRoBa. ISRobotNet has adapted and extended MeRMaID (Multiple Robot Middleware for Intelligent Decision-making), into the YARP networking software, that forms the basis middleware over which a wide variety of integration architectures can be deployed. Finally, CONET-IT uses Player, a set of open-source software tools for robot and sensor applications. Player module manages the WSN messages that should comply with the WSN-Player Driver interface. TinyOS2 is recommended, and the Contiki compliancy is under final tests. Any program can communicate with Player over a TCP/IP socket and nodes are USB-plugged to their carrier robot. The same goes for IoT-LAB since the sensor can communicate with its carrier robot, enabling cooperative mobility where the robot follows sensor instructions. Furthermore, on both testbeds, full control of the robot is given to the user, enabling experimentations both in the wireless sensor network area and in the robotic area.

Repeatability of experiments can be provided by means of mobility models. Using the trains provided by IoT-LAB, it is easy to repeat the exact same experiment since trains are always moving on the same circuit. However, when mobility is more complex, mobility patterns should be provided, helping users in the design of their experimentations. RoombaNet has implemented a "semi-random" model where the experimenter can modify the level of randomness in the pattern. On MOTEL and IoT-LAB testbeds, experimenters are able to load their predefined paths for the robots or to modify the paths by interactively moving, adding or deleting waypoints. This

offers an efficient and quick repeatability of the experiments. CONET-IT and IoT-LAB also handle obstacles avoidance.

As previously mentioned in the mobility requirements (Section II-D), in such a context, it is essential to have visualisation tools available, to study the dynamics of the experiments in a testbed. Therefore, MINT-m's MOVIE interface is a very complete tool that allows the collection of the nodes path traces, real-time display of network traffic load distribution as well as position of individual nodes and inter-node signal-to-noise ratios. MOTEL's Muroba tool allows the visualisation of the real locations of the robots throughout the experiment, and acts as a graphical post-experiment analysis tool. Since it is a remotely accessible testbed, CONET-IT users need to have a vision on the physical environment and the running experiment. In that purpose, CONET-IT provides very efficient integrated testbed GUI, described in Section III-A.

IV. Concluding discussions

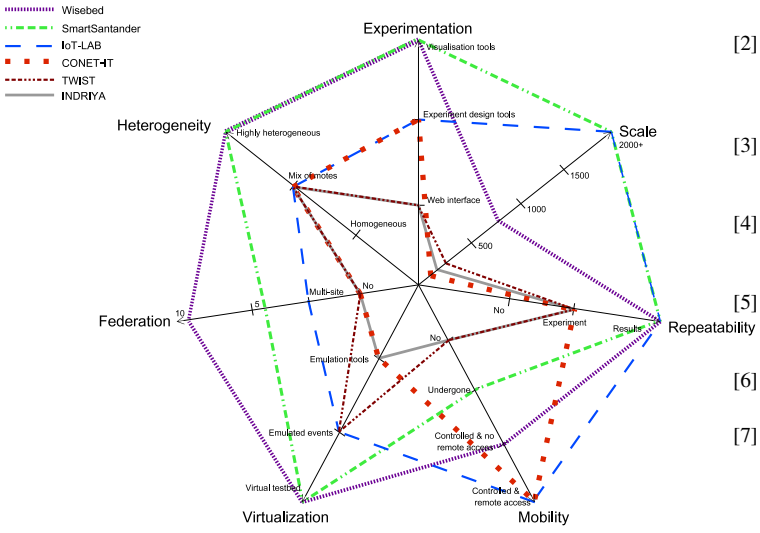

Fig. 1. Overview of strengths and weaknesses of surveyed testbeds

Many platform initiatives have pop up to help the design of IoT solutions. Designing, deploying, setting up and then maintaining such platforms is a very difficult task as testifies the fact that large number of used testbeds have been decommissioned (MoteLab, TrueMobile). This survey describes some of the most currently active and meaningful IoT testbeds. Figure 1 features their different characteristics at a glance. It shows that testbeds are more and more heterogeneous and multi-purpose. They offer an increasing set of services to answer a large range of user expectations, even though none of them manages to fulfill all requirements, especially in terms of maintenance. We can note that SmartSantander and WISEBED are the two answering the best on criteria we listed. Even if IoT-LAB is slightly set back, it is strongly oriented to scale and repeatability criteria, while answering at the same time to mobility concerns. If the user focuses his work on mobility only, CONET-IT could the better choice regarding experimentation means. Surviving testbeds are wonderful tools for IoT developments. Most of the ones discussed in this article follow the technological trends and continuously open new services in order to integrate societal behaviour modifications and arising challenges. Some of the potential future directions to be integrated are the inclusion of actuators to allow the user to act on the environnement and change some of its characteristics by opening doors, turning lights on, etc. Also, new hardware should feature some energy harvesting components (solar cell, piezo cells, etc) and new tools should be designed to ease the tests for security and safety research purposes. Another direction to extend and improve testbeds could be to introduce the RFID technology into the experiments, thus making possible automatic identification and tags tracking.

To conclude, this survey shows the great advances made by IoT experimental platforms in different directions, offering a large set of diversified services. They all present some strengths and lacks. We expect this survey assist a potential user to easily choose regarding his own needs.

References

[1] M. Nati, A. Gluhak, H. Abangar, and W. Headley, "Smartcampus: A user-centric testbed for internet of things experimentation," in *Proc. of WPMC*, Atlantic City, USA, 2013.

[2] C. B. des Roziers, G. Chelius, T. Ducrocq, E. Fleury, A. Fraboulet, A. Gallais, N. Mitton, T. Noël, and J. Vandaele, "Using senslab as a first class scientific tool for large scale wireless sensor network experiments," in *Proc. of Networking*, Valencia, Spain, 2011.

[3] A. Jimenez-Gonzalez, J. R. M.-D. Dios, and A. Ollero, "An integrated testbed for cooperative perception with heterogeneous mobile and static sensors," *Sensors*, vol. 12, 2011.

[4] V. Handziski, A. Köpke, A. Willig, and A. Wolisz, "TWIST: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks," in *Proc. of REALMAN*, Florence, Italy, 2006.

[5] A. Arora, E. Ertin, R. Ramnath, M. Nesterenko, and W. Leal, "Kansei: A high-fidelity sensing testbed," *IEEE Internet Computing*, vol. 10, pp. 35–47, 2006.

[6] D. Sakamuri and H. Zhang, "Elements of sensornet testbed design," in *Handbook of Sensor Networks*, 2009, ch. 35, pp. 1–36.

[7] G. Coulson, B. Porter, I. Chatzigiannakis, C. Koninis, S. Fischer, D. Pfisterer, D. Bimschas, T. Braun, P. Hurni, M. Anwander, G. Wagenknecht, S. Fekete, A. Kröller, and T. Baumgartner, "Flexible experimentation in wireless sensor networks," *ACM Communications*, vol. 55, no. 1, pp. 82–90, Jan. 2012.

[8] O. Hahm, M. Güneş, and K. Schleiser, "Des-testbed a wireless multi-hop network testbed for future mobile networks," in *Proc. of KuVS*, Hannover, Germany, 2010.

[9] R. Lim, F. Ferrari, M. Zimmerling, C. Walser, P. Sommer, and J. Beutel, "Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems," in *Proc. of IPSN*, Philadelphia, USA, 2013.

[10] M. Doddavenkatappa, M. Chan, and A. Ananda, "Indriya: A low-cost, 3d wireless sensor network testbed," in *TridentCom*, Thessanoliki, Greece, 2012.

[11] S. Bouckaert, W. Vandenberghe, B. Jooris, I. Moerman, and P. Demeester, "The w-ilab.t testbed," in *TridentCom*, Shanghai, China, 2011.

[12] M. A. Hail, J. Pinkowski, T. Teubler, M. Danckwardt, D. Pfisterer, and H. Hellbrück, "Roombanet - testbed for mobile networks." *Electronic Communications of the EASST*, vol. 37, 2011.

[13] E. Förster, A. Förster, T. Leidi, K. Garg, D. Puccinelli, F. Ducatelle, S. Giordano, and L. M. Gambardella, "Poster abstract: Motel: Towards flexible mobile wireless sensor network testbeds," in *Proc. of EWSN*, Bonn, Germany, 2011.

[14] P. De, A. Raniwala, R. Krishnan, K. Tatavarthi, J. Modi, N. Syed, S.Sharma, and T. Chiueh, "MiNT-m: An autonomous mobile wireless experimentation platform," in *Proc. of Mobisys*, Uppsala, Sweden, 2006.

[15] M. Barbosa, A. Bernardino, D. Figueira, J. Gaspar, N. Gonçalves, P. U. Lima, P. Moreno, A. Pahliani, J. Santos-Victor, M. T. J. Spaan, and J. Sequeira, "ISRobotNet: A testbed for sensor and robot network systems," in *Proc. of IROS*, St. Louis, USA, 2009.