Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080301

Wireless Sensor Networks Testbeds and State-of-the-Art Multimedia Sensor Nodes

Muhammad Omer Farooq[1,∗] *and Thomas Kunz*[2]

[1] Institute of Telematics, University of Luebeck, Luebeck, Germany
[2] Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada

Received: 3 May. 2013, Revised: 7 Sep. 2013, Accepted: 8 Sep. 2013
Published online: 1 May. 2014

Abstract: In recent years, Wireless Sensor Networks (WSNs) attracted researchers' attention. This has led to advancements made in both hardware design and software components. The research at the hardware level has resulted in a variant of WSN called Wireless Multimedia Sensor Networks (WMSNs). WSNs have application in battlefields, industrial process monitoring, and environmental monitoring, to name but a few. Furthermore, WMSNs have the potential for real-time object detection and recognition. It can be used to locate misplaced items, assisted living, helping people with cognitive impairments, species detection, and providing real-time images for many other surveillance-based applications. Thus, such networks need to operate in a diverse environment. Researchers invariably test their designed algorithms and protocols using simulation tools. Simulators do not accurately model the environment in which such networks are deployed. Therefore, there exists many WSNs and WMSNs testbeds. These testbeds enable researchers and programmers to validate the performance of their algorithms and protocols on a physical network. In this paper, we provide a detailed description of various state-of-the-art WSN testbeds. We have evaluated these state-of-the-art testbeds on a set of metrics. Furthermore, we survey the state-of-the-art in wireless multimedia sensor nodes hardware along with their eminent features.

Keywords: Wireless Sensor Networks (WSNs), Wireless Multimedia Sensor Networks (WMSNs), Wireless Sensor Networks Testbeds, Wireless Multimedia Sensor Nodes

1. Introduction

Recent years have seen tremendous research activity in the WSN domain. Research in WSNs is resulting in a multitude of new protocols and systems. Invariably, scientists use simulation tools and mathematical modeling to evaluate their proposed protocols. Since WSNs have the potential to operate in diverse environments, simulation tools and mathematically modeling do not capture physical channel characteristics accurately. This results in in-accurate performance evaluations. Therefore, some researchers have focused on designing WSN testbeds, so that system and protocol performance can be evaluated in a more realistic setting. Recently research has also resulted in a variant of WSN called WMSNs. Audio/video sensing devices constitute a WMSN. These devices possess capabilities of capturing, processing, and transmitting multimedia data data: audio, video, and stationary images. Moreover, advancement made in hardware design, coding theory, signal processing, and wireless networking have made WMSNs a reality and a multidisciplinary research field.

Ordinary sensor nodes cannot capture and process multimedia data. Therefore, manufactures and researchers provide solutions like Cyclops [1], Stargate [2] and Imote2 [2]. To evaluate WMSNs systems, researchers have started to design and deploy WMSN testbeds, for same reasons WSN testbeds have become popular. There exist a range of WMSNs testbed, details can be found in [3].

We have organized this paper as follows. State-of-the-art hardware for WMSNs is analyzed in Section 2. Section 3 presents a detailed description of WSNs testbeds. Finally, we conclude this paper in Section 4.

2. State-of-the-Art WMSN Nodes

In this section, we briefly summarize the state-of-the-art in WMSN nodes.

[∗] Corresponding author e-mail: farooq@itm.uni-luebeck.de

2.1. Stargate

Stargate [2] uses an Intel PXA-255 Xscale processor that operates at 400 MHz. It has 64 MB of SDRAM and 32 MB of flash storage. Its processor can execute advanced image processing algorithms. It supports the embedded LINUX (kernel 2.4.19) operating system. It supports two radios i.e., the IEEE 802.11 and the IEEE 802.15.4.

2.2. Imote2

Imote2 [2] is an advanced platform especially designed for sensor network applications requiring high CPU/DSP and wireless link performance and reliability. The Imote2 contains an Intel XScale processor, PXA271. It comes with an IEEE 802.15.4 radio (TI CC2420) with an onboard antenna. It has two basic sensor board interfaces, consisting of two connectors. Moreover, it has an advanced sensor board interface, consisting of two high density connectors on the other side of the board. It has 256 KB SRAM, 32 MB Flash, and 32 MB SDRAM and an integrated 2.4 GHz antenna.

2.3. CMUCam3

The CMUCam3 [4] is an ARM7TDMI based fully programmable embedded computer sensor. It is equipped with a Philips LPC2106 processor that is connected to an Omni-vision CMOS camera sensor module. Features of CMUCam3 include: CIF resolution (352 x 288 pixels) RGB color sensor, MMC flash slot with FAT16 driver support, loads images into memory at 26 frames/sec. It contains LUA (a lightweight scripting programming language) that allows for rapid prototyping, software-based JPEG compression and a bcmucam3asic image manipulation library. It contains 64 KB RAM and 128 KB of ROM.

2.4. MeshEye

MeshEye [5] is a hybrid resolution smart camera node for applications in distributed intelligence surveillance. MeshEye has a unique vision system i.e., a low resolution stereo vision system continuously determines position, range, and size of moving objects entering in its field of view. This information triggers a color camera module to acquire a high resolution image of the object. It contains an ARM7TDMI- ARM thumb processor. It is a 32 bit RISC architecture that can operate at up to 47.92 MHz. MeshEye has 64 KB of SRAM and 256 KB of flash. It uses a CC2420 transceiver and provides an implementation of the IEEE 802.15.4 standard.

2.5. WiCa

WiCa [6] is a smart camera node with high performance vision system. It uses two VGA camera modules (640 x 480) 24 bit color, which feed video to IC3D, which is a dedicated parallel processor, running at 80 MHz. WiCa uses an 8051 based Atmel AT89C51 host processor, which runs at 24 MHz, through a 128 KB dual port RAM. For large scale image processing it supports 10 Mbits of RAM. It provides an implementation of the IEEE 802.15.4 standard.

2.6. Cyclops

Cyclops [1] is a small camera device developed for WMSNs. Cyclops can be interfaced with MICA2 and MICAz nodes. The Cyclops hardware architecture consists of an Agilent ADCM-1700 CMOS camera module, a Xilinx FPGA and an 8 bit RISC ATMega128 micro-controller.

2.7. Summary

There exist a range of wireless multimedia sensing nodes. For transmitting multimedia data over low bandwidth links these nodes have capabilities to run advanced coding algorithms. Furthermore, sufficient resources in terms of memory and processing capabilities are available to run advanced image processing algorithms.

3. Wireless Sensor Networks Testbeds

In this section, we elaborate on the state-of-the-art WSN testbeds. We evaluate various existing WSN testbeds based on metrics such as: number of deployed nodes, heterogeneity in terms of supported hardware and software, availability to the general public, and deployment scale.

3.1. WISBED

The WISEBED [12] is a large-scale WSN testbed and it is a joint effort of nine European Universities and Research Institutes. The main goals of WISEBED are: heterogeneous WSN testbed, testbed virtualization, facilitate end users and application through variety of interfaces, and unified algorithmic and software environment.

WISEBED follows a hierarchical architecture that revolves around four main entities: wireless sensor nodes, gateways, portal server, and overlay network. Wireless sensor nodes comprise the WSN and they are at the lowest level of the hierarchy. A set of wireless sensor

Table 1: WISEBED Summary

University/Institute	Total Nodes	Node Type	Micro-processor	Radio	RAM/FLASH
University of Luebeck	162	iSense [7]	Jennic 32-bit (JN5139R1 and JN5148)	IEEE 802.15.4	128KB/512KB
		TelosB [8]	Texas Instruments 16-bit (TI MSP 43)	IEEE 802.15.4	10KB/1MB
Freie University	100	DES-node [9]	AMD Geode LX800 and ARM7	IEEE 802.11a/b/g	98KB/512KB
Braunschweig Institute of Technology	30	iSense [7]	Jennic 32-bit (JN5139R1)	IEEE 802.15.4	128KB/512KB
Computer Technology Institute	154	iSense [7]	Jennic 32-bit (JN5139R1 and JN5148)	IEEE 802.15.4	128KB/512KB
		TelosB [8]	Texas Instruments 16-bit (TI MSP 43)	IEEE 802.15.4	10KB/1MB
Technical University Catalonia	12	iSense [7]	Jennic 32-bit (JN5139R1)	IEEE 802.15.4	128KB/512KB
University of Bern	47	MSB-430 [10]	(TIMSP430F1612)	Chipcon CC1020	5KB/55KB
		TelosB [8]	Texas Instruments 16-bit (TI MSP 43)	IEEE 802.15.4	10KB/1MB
University of Geneva	50	iSense [7]	Jennic 32-bit (JN5139R1)	IEEE 802.15.4	128KB/512KB
Delft University of Technology	140	G-node [11]	Texas Instruments (MSP430F2418)	TI CC1101	8KB/8Mb
		T-node [11]	Atmel Atmega 128L	TI CC1101	4KB/4Mb
		Tmote Sky [2]	Texas Instruments (MSP430F2418)	TI CC2420	10KB/8Mb
Lancaster University	16	TelosB [8]	Texas Instruments 16-bit (TI MSP 43)	IEEE 802.15.4	10KB/1MB

nodes is connected to a gateway that allows access to the attached sensor nodes. The gateways are connected to a portal server that not only manages the WSN but also enables user interaction with the testbed. Each WISBED site maintains a separate portal server. The portal servers at different WISBED sites are interconnected to form an overlay network that can provide virtual access to each connected testbed located at geographically different locations. Portal servers provide services to enable communication among different portal servers so that the goal of testbed virtualization can be fulfilled.

In order to accomplish the heterogeneous WSN testbed goal, the testbed deployed at different sites contain wireless sensor nodes from different vendors that differ in capabilities and hardware. Furthermore, it is possible that each testbed supports different software/firmware but the middleware makes it possible to keep interoperability among these testbeds.

The operating systems supported by the testbed are Contiki [13], iSense [7], and TinyOS [14]. The WISEBED testbed provides a C++ library called WISELIB, which contains implementation of different WSN related algorithms, moreover applications written using WISELIB can be compiled to any of the supported operating systems. In order to evaluate the performance of different algorithms for WSN, the WISEBED testbed provides a WSN simulator called Shawn.

WISEBED is an open testbed. A user can access the testbed by creating an account at any of the WISEBED's site, afterwards the user can freely use the WSN testbed. To access the testbed, WISEBED provides three methods: a web-based API, web-based clients, and desktop-based clients. Hence, user friendliness is achieved through supporting different hardware/software platforms along with a range of methods to access the testbed. Table 1 summaries WISEBED.

3.2. SensLAB

SensLAB [15] is a very large-scale WSN testbed with a total of 1000 nodes deployed at four sites in France. The main goal of SensLAB is to offer an accurate and efficient scientific tool to help in the design and development of large scale WSN. SensLAB is currently deployed at: (i) INRIA Grenoble, (ii) INRIA Lille, (iii) INRIA Rennes, and (iv) University of Strasbourg.

The hardware used in the SensLAB testbed includes different versions of WSN430 nodes [15]. It includes WSN430 heart rate daughter board, WSN430 GPS-Accelerometer daughter board, WSN430 testbed daughter board, WSN430 daughter board, WSN430 daughter board, WSN430 strain daughter board, WSN430 Bluetooth daughter board, and WSN430 motion capture daughter board.

From the software perspective SensLAB provides support for three operating systems: Contiki [13], FreeRTOS [16], and TinyOS [14]. SensLAB provides a software programming library that includes implementation of various OS free Medium Access Control (MAC) protocols i.e., Carrier Sense Multiple Access (CSMA), Time Division Multiple Access (TDMA), and XMAC. Furthermore, it provides an implementation of a localization protocol and a gradient-based routing algorithm. SensLAB provides support for WSN and embedded system simulations in the form of WSNet [15], and Wsim [15]. SensLAB is open to researchers of the host institutes and outside users need to request permission in order to use the testbed. SensLAB provides a web-based system to access the testbed.

3.3. MoteLab

MoteLab [17] is a WSN testbed developed at the Electrical and Computer Engineering department of

Table 2: WSN Testbeds Comaprison

Testbed	No. of Nodes	Hardware Heterogeneity	Software Heterogeneity	Availability	Deployment Scale
WISEBED [12]	711	Yes	Supports multiple operating systems, provides implementation of network simulator, and software API library for application development	Public	Developed in multiple countries of Europe
SensLAB [15]	1000	No	Supports multiple operating systems, provides implementation of network simulator, and a software programming library	Through request	Four locations in France
moteLab [17]	190	Only Tmote	Only TinyOS	Public	Maxwell Dworkin Laboratory, Harvard University
CitySense [18]	100	No	Linux-based	Public	City-wide
Sensei [19]	20	Only TelosB	TinyOS and Contiki	Public	Lab. Level deployment at Uppsala University

Harvard University. It is a public testbed, users can run their WSN applications using a web-based interface. Registered users can schedule their applications on wireless sensor nodes, upload binary images, and visualize wireless sensor nodes output through the web-based interface. The testbed stores the output of wireless sensor nodes in a database and the detailed output is presented to the user once the job is completed.

The design of the MoteLab testbed revolves around the server, a MySQL database backend, a web interface, a DBLogger, and a Job Daemon. The server hosts the web interface, different databases, and provides connectivity to the wireless sensor nodes. The web interface enables the registered users to create a new job i.e., to upload binary images, and to assign binary images to different wireless sensor nodes. Moreover, the web interface can be used to modify or delete already created jobs. The DBLogger stores the messages corresponding to the active job(s) and on the completion of the job(s) the messages are presented to the user. The job daemon is responsible for reprogramming the nodes, killing processes pertaining to finished jobs, and allocating and de-allocating other system components.

Moreover, the MoteLab testbed imposes a quota on each registered user in terms of total duration of pending jobs i.e., if the user quota allows 40 minutes of testbed usage then the user cannot run jobs that exceed the 40 minutes duration.

MoteLab only supports the TinyOS operating system, therefore users need to know the NesC programming language. Secondly, the MoteLab team does not provide a software-based API to facilitate the application development process on the MoteLab testbed.

The MoteLab testbed consists of 190 Tmote Sky [2] wireless sensor nodes. The wireless sensor node specifications are a TI MSP430 processor, 10 KB RAM, 1Mb flash, and Chipcon CC2420 radio. Each node is also connected to the Ethernet.

3.4. CitySense

CitySense [18] is an open wireless mesh and sensor networking testbed that spans the entire city of Cambridge, Massachusetts, USA. The goals of CitySense include: an evaluation platform for WSN applications, support for new mesh routing algorithms, sensor networking at the urban-scale, new distributed algorithms for in-network data processing and aggregation, novel programming abstractions, and to enable users to reprogram the nodes using the Internet.

The key features of CitySense are: city-wide deployment and monitoring of the physical world through sensors. The CitySense aim was to deploy a city-wide wireless mesh and sensor network testbed. Therefore, it was not feasible to use sensor nodes that communicate at a range of less than 100 meters (primarily due to the cost factor). Furthermore, the CitySense team envisions that future WSN applications will demand high data rate and complex processing at the nodes. Therefore, they decided to develop sensor nodes that can communicate using the IEEE 802.11 standard. This meets the high data rate requirements and reduces the cost of deployment.

The CitySense testbed includes wireless sensors nodes, wire-line gateways, back-end servers, and a web-based interface. The wire-line gateways link the wireless mesh to the Internet and to the back-end servers. The server hosts databases pertaining to the testbed, registered users, and users experimental data. Furthermore, the web interface is hosted on the server that enables users interaction with the testbed. The wireless nodes are mounted on streetlights and are powered through these streetlights as well.

3.5. Sensei

Sensei [19] is a nomadic WSN testbed developed at Uppsala University. The distinguishing feature of Sensei is that it provides support for mobile nodes and its nomadic nature makes it possible to evaluate WSN application performance in a range of environments.

Wireless sensor nodes, sensor hosts, site manager, and monitors constitute the design of the Sensei testbed. A group of sensor nodes is connected to a sensor host and there are a number of sensor hosts present in the testbed. Wireless sensor nodes connect to the sensor host through the USB interface. Since the testbed supports mobile nodes, sensor hosts keep track of mobile nodes and inform the site manager about the location of these nodes periodically. Communication among sensor hosts takes place using the IEEE802.11 standard. For communication between sensor hosts and the site manger, a control channel is setup using the IEEE802.11 standard. The site manager controls the WSN testbed. The site manger has a wire-line connection with the monitor. The site manager acts as the gateway to the testbed and logs events. The monitor presents the output of the sensor nodes along with other control information to the users. A Java-based desktop client is used to enable users to interact with the testbed.

The testbed uses a Linux based Asus WL-500G wireless access point as sensor hosts and TelosB [8] sensor nodes with CC2420 radio transceiver. The testbed provides support for both TinyOS and Contiki operating systems.

4. Conclusion

There exist a range of powerful multimedia sensing nodes. A lot of effort was made in past few years in setting up large, shared, heterogeneous WSN testbeds. Little work yet to integrate these new multimedia sensors into these testbeds. The latter is necessary as new nodal capabilities require new protocols and pose challenges: Quality of Service, transmitting multimedia data over low bandwidth links, and memory management to name but a few. There is an opportunity for new types of applications that require new information coding techniques, network resource management, and algorithms and protocols to transfer multimedia data over low bandwidth links.

References

[1] M. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, E. Estrin, and M. Srivastav. Cyclops: in situ Image Sensing and Interpretation inWireless Sensor Networks. *In the proc. of the 3rd International Conference on Embedded Network Sensor Systems*, (2005).

[2] MEMSIC. [online], http://www.memsic.com..

[3] M. O. Farooq and T. Kunz. Wireless Multimedia Sensor Networks Testbeds and State-of-the-Art Hardware: A Survey. *In FGIT-FGCN*, **1**, 114 (2011).

[4] A. Rowe, A. Goode, G. Dhiraj, and N. Illah. CMUcam3: An Open Programmable Embedded Vision Sensor. Technical report, Carnegie Mellon Robotics Institute, (2007).

[5] S. Hengstler, D. Prashanth, S. Fong, and H. Aghajan. MeshEye: A Hybrid-Resolution Smart Camera Mote for Application in Distributed Intelligent Surveillance. *In the proc. of the 6th International Symposium on Information Processing in Sensor Networks*, (2007).

[6] R. Kleihorst, A. Abbo, B. Schueler, and A. Danilin. Camera mote with High-Performance Parallel Processor for Real-Time Frame-based Video Processing. *In the proc. of the IEEE Conference on Advanced Video and Signal based Surveillance*, (2007).

[7] iSense. [online], http://www.coalesenses.com/index.php?page=software-system.

[8] TelosB. [online], http://www.willow.co.uk/html/telosbmoteplatform.html.

[9] DES-nodes. [online], http://www.des-testbed.net/sites/default/files/DES-Testbed-Information_Flyer-June-2010.pdf

[10] Modular-Sensor-Board-430. [online], http://cst.mi.fu-berlin.de/Projects/ScatterWeb/modMSB-430.html.

[11] SWONet-Technologies. [online], http://www.sownet.nl/index.php/products.

[12] I. Chatzigiannakis, C. Koninis, G. Mylonas, S. Fischer, and D. Pfister. WISEBED: an Open Large-scale Wireless Sensor Network Testbed. *In the proc. of the 1st International Conference on Sensor Network Applications, Experimentation, and Logistics*, (2009).

[13] A. Dunkels, B. Gronvall, and T. Vogit. Contiki A Lightweight and Flexible Operating System for Tiny Networked Sensors. *In the proc. of the 29th Annual International Conference on Local Computer Networks*, (2004).

[14] P. Levis, S. Madden, J. Polastre, R. Szewcyzk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler. TinyOS: An Operating System for Sensor Networks, http://dx.doi.org/10.1007/3-540-27139-27.

[15] Senslab. [online],http://www.senslab.info.

[16] FreeRTOS. [online], http://www.freertos.org.

[17] G. W. Allen, P. Swieskowsli, and M. Welsh. MoteLab: A Wireless Sensor Network Testbed. *In the proc. of the 4th International Symposium on Information Processing in Sensor Networks*, (2005).

[18] G.W. Allen, P. Swieskowsli, and M. Welsh. CitySense: An Urban-Scale Wireless Sensor Network and Testbed. *In the proc. of the IEEE Conference on Technologies for Homeland Security*, (2008).

[19] O. Rensfelt, F. Hermans, C. Ferm, P. Gunningberg, and L.A. Larzon. Sensei-UU: A Nomadic Sensor Network Testbed Supporting Mobile Nodes. *In the proc. of the 4th ACM International Workshop on Wireless Networks Testbeds, Experimental Evaluation, and Characterization*, (2010).

Muhammad Omer Farooq has received his BS (Computer Science) and MS (Computer Engineering) degrees with distinction from the Virtual University of Pakistan and the Center for Advance Studies in Engineering (CASE) Islamabad, Pakistan respectively. He is a PhD student in the Institute of Telematics, University of Luebeck, Germany. His research focuses on algorithms and protocols for wired and wireless computer networks. In recent years, his research has focused on QoS provisioning in the Internet, Mobile Ad-hoc Networks, and Wireless Multimedia Sensor Networks.

Thomas Kunz is currently a professor in the Department of Systems and Computer Engineering at Carleton University, Ottawa, Canada. He heads the Mobile Computing Group, researching wireless network architectures (manets, wireless mesh networks, and wireless sensor networks), network protocols (routing, mobile IP, and QoS support), and middleware layer for innovative wireless applications. He has served on more than 50 TPCs of international conferences and workshops in the mobile and wireless domain. He is author or coauthor of more than 190 fully reviewed technical papers and received a number of awards and best paper prizes. He is a senior member of both IEEE and ACM.