Computers, Environment and Urban Systems 65 (2017) 41–52

Contents lists available at ScienceDirect

Computers, Environment and Urban Systems

journal homepage: www.elsevier.com/locate/ceus

Enhancing reliability in Wireless Sensor Networks for adaptive river monitoring systems: Reflections on their long-term deployment in Brazil

Jó Ueyama[a,d,*], Bruno S. Faiçal[a,b], Leandro Y. Mano[a], Guilherme Bayer[a], Gustavo Pessin[c], Pedro H. Gomes[d]

[a] *University of São Paulo (USP), São Carlos, São Paulo, Brazil*
[b] *State University of Northern Paraná (UENP), Bandeirantes, Paraná, Brazil*
[c] *Instituto Tecnológico Vale (ITV), Belém, Pará, Brazil*
[d] *University of Southern California (USC), Los Angeles, CA, USA*

ARTICLE INFO

Article history:
Received 22 July 2016
Received in revised form 11 April 2017
Accepted 15 May 2017
Available online 22 May 2017

Keywords:
Wireless Sensor Network
Smart cities
Sensing critical environments
Public safety
Disaster resilience
Communications networks

ABSTRACT

Several adaptive systems have been proposed that are based on the concepts of smart cities, which can be successfully adapted to natural disasters or other public safety concerns. Since these systems are embedded in a critical and dynamic environment, it is really important to have an infrastructure that is capable of providing real-time environmental information. This paper discusses two research questions that arise from adaptive ubicomp systems: (*i*) *what* are the key requirements to provide a reliable WSN-based system (e.g. a river monitoring system)? and (*ii*) *how* can an adaptable and reliable WSN-based system be developed? This paper seeks to respond to the former question with the aid of the RESS standard platform. The latter question is answered by employing a generic approach for adaptation. The term "critical systems", means that any error may result in the loss of human life. We devised the RESS standard after deploying the WSN-based river monitoring system in Brazil for five years. Our prototype underwent several trials, sometimes leading to failure or damage, before we came up with a more reliable solution, which is outlined in this article. Finally, while our RESS platform is policy-free, it is extensible/adaptable and hence can naturally be adapted to new policies.

1. Introduction

More than half of the world's population is currently located in urban areas, with a concentration of up to about 80.2% in countries with higher income levels, and forecasts suggest that this proportion will increase (United Nations, 2014). The unbridled growth of cities has caused an increase in the number of natural disasters and the use of technology to overcome these problems has been investigated by many researchers (Araujo, Duarte-Figueiredo, Tostes, & Loureiro, 2014; Neto, Guidoni, & Villas, 2014; Ueyama et al., 2014). These studies adhere closely to the concept of smart cities, which allows the resources and services of an urban city to be managed in an optimal manner (Komninos, 2011).

Wireless Sensor Network (WSN) have been used in a large number of applications that range from health-care, structural monitoring, asset tracking to environment monitoring and disaster and emergency response (Rashid & Rehmani, 2016; Rehmani & Pathan, 2016; Yick, Mukherjee, & Ghosal, 2008). In the particular case of disasters and emergency response, WSNs are used to monitor natural phenomena, such as storms, hurricanes or volcanos and supply computing systems with information to assist in decision-making (Baggio, 2005; Kim, Jabro, & Evans, 2011; Ueyama et al., 2014). One example of an application that is exploited in this paper is rain monitoring in urban areas that are prone to flooding.

It should be pointed out that many WSN applications are not affected by single-supply sensing error at any given time (such as failing to measure a data reading used to generate an average). In contrast, in environments that are dynamically critical (like those addressed in this paper) the reliability of the WSN is a factor of extreme importance. In these scenarios there is a need for information to be available in real time so that it can be used for immediate action. In addition, even small errors may result from the unavailability of information, which may cause systems or sectors that use this information to make wrong decisions, and increase the odds of the loss of human life.

* Corresponding author at: University of São Paulo, Av. Trab. São-Carlense, 400, São Carlos - SP, 13566-590, Brazil.
E-mail addresses: joueyama@icmc.usp.br (J. Ueyama), bsfaical@uenp.edu.br (B.S. Faiçal), leandroyukiomano@usp.br (L.Y. Mano), guilherme.bayer@usp.br (G. Bayer), gustavo.pessin@itv.org (G. Pessin), pdasilva@usc.edu (P.H. Gomes).

http://dx.doi.org/10.1016/j.compenvurbsys.2017.05.001

With this in mind, this paper discusses two research questions: (*i*) *what* are the key factors to take into account while striving to provide a reliable WSN-based critical systems, such as a flood monitoring system? and (*ii*) *how* can we develop an adaptable and reliable WSN-based critical system? The first question is answered by means of our proposed RESS standard platform. The second is answered by employing our generic approach for adaptation. We devised a reliable WSN/IoT-based river monitoring system that is policy-free and accepts new policies in a plug & play fashion. Our system has been successfully deployed in the city of São Carlos/Brazil and has been running for over five years. Our approach has been adopted for several systems, including mobile phone software and follows our generic approach for adaptation.

The rest of the paper is structured as follows. Section 2 outlines the existing guidelines for WSN. Section 3 provides an overview of our current WSN implementation used for river monitoring and examines its long-term deployment. Our proposal is set out in Section 4. In Section 5, there is an explanation about how the data on the behavior of the sensor nodes were gathered and analyzed. Finally, in Section 6 we conclude this work, and point out areas that need to be explored in further studies.

2. Existing guidelines for implementing a WSN

Since WSNs are widely employed for monitoring environments with different features, works can be found in the literature that set out guidelines for implementing this technology. These studies address the challenges raised by the real deployment of WSNs (e.g. uninterrupted monitoring under stochastic conditions in different environments).

Gomez, Laube, and Sorniotti (2008) highlight the importance of WSNs in different fields, such as defense, healthcare, and traffic control. Furthermore, their ability to monitor the physical environment has led to their use in industrial and commercial applications, where information is obtained and processed in accordance with the needs of customers. The paper discusses the design guidelines that have been drawn up to integrate WSNs with Enterprise Systems (Enterprise Resource Planning System - ERP System). This involves combining a top-down approach of context-aware middleware with the bottom-up approach of WSNs middleware, for the design of a SOA-based architecture. This architecture provides a standardized, secure and reliable way for companies to use processed and/or raw data acquired by the sensor nodes for their business purposes.

However, the guidelines relied on by Gomez et al. (2008) fail to look at the implementation of WSNs from an architectural standpoint, but instead the WSNs are implemented in accordance with business regulations, to ensure they meet the conditions required to be integrated with Enterprise Systems. Thus, it can be assumed that the architecture implemented for the WSN is robust enough to prevent any failure during the operation. In addition, this study does not take into account the complexity of the environment to be monitored, although these features exert a strong influence on the architectural implementation of WSN projects.

In Giannopoulos, Goumopoulos, and Kameas (2009), there is a description of the design of a WSN to monitor environmental variables such as temperature, humidity, light and soil moisture. Their paper shows how to deal with problems that arise during the implementation of a real WSN, such as network synchronization, data consistency, data aggregation and energy saving. On the basis of their experience it is possible to draw up some guidelines for building similar systems, which share the objective of monitoring environments for long periods.

However, the experience of Giannopoulos et al. (2009) was based on the implementation of two sensor nodes in a controlled environment (a laboratory) and the expansion of the network (which involved more nodes) and the analysis was only conducted in a simulated way. Thus, although this was a valuable study, the difficulties and obstacles of implementing a WSN in a real-world environment are not addressed, since in real environments there are countless external factors that can influence the design of the system.

Mampentzidou, Karapistoli, and Economides (2012) prepared a set of guidelines to implement WSNs in agriculture, particularly in applications for monitoring crops. First, the authors categorize some key applications of WSN in Precision Agriculture - PA and summarize the components used in the WSN. Following this, there is a description of, the general guidelines that show how to deploy a WSN for agricultural applications. The paper makes a significant contribution to the field by addressing questions regarding hardware and software, communication, sensing, power supply, maintenance, and the importance of interacting with farmers.

However, there was a lack of a dynamic or critical environment for the proposed guidelines. Moreover, it is not essential to sense a possible disaster (which depends entirely on non-deterministic factors) to act as a monitoring platform for adaptive systems. Thus, the proposal does not aim at keeping the WSN working when there are natural disasters; in these situations, it is acceptable for failures to occur as a result of these phenomena.

The guidelines set out by Mampentzidou et al. (2012) cannot be regarded as appropriate for a WSN when they are implemented in critical environments. These scenarios – clarification needed here – are cited in future work, where the authors state that the proposed guidelines can be improved to provide support for other applications such as the environmental monitoring of natural disasters (e.g., volcanoes, landslides, wild fires).

With regard to the use of WSN technology for flood monitoring, there are several works that are worth citing. These include the work in Castillo-Effer, Quintela, Moreno, Jordan, and Westhoff (2004) that adopts WSNs for flash-flood warnings. This work was conducted in the Andean region of Venezuela and was designed to monitor the environment and track the disaster while it was taking place. However, the project was not concerned with assessing the nature of the environment, but rather in how monitoring was carried out with WSNs. Moreover, the aim of the experiment was not to provide an adaptive and tailorable WSN-based system.

Another similar work called RTFMS (a Real-Time Flood Monitoring System) (Sunkpho & Ootamakorn, 2011) was conducted in the island of Mauritius. This is also WSN-based and involves carrying out a real deployment in the field. The main value of this work is that it deals with forecasting and not only detection. It incurs small overhead in real-time, and is able to estimate the extent of the flooding very fast. The work made use of SunSPOT and MicaZ technologies, which are not very robust. Moreover, it does not include an evaluation for a long-term deployment and again is not designed to show how a reliable and runtime adaptable WSN-based system can be built.

Finally, the work in Seal et al. (2012) is also a WSN-based system that deals with flood monitoring. Similar to the work in Sunkpho and Ootamakorn (2011), it is mainly concerned with providing a model for predicting floods. This study has a prediction model based on multiple robust linear regressions. However, the authors are more concerned with displaying the algorithm itself rather than the WSN system. In other words, it is theoretical rather than an attempt to examine the system itself. Hence, this work does not include a field test and can only operate in simulated environments. As a result, it does not target long-term evaluation through a real deployment in the field.

The papers cited in this section represent the state-of-the-art recommendations for the implementation of WSNs in real applications. It should be noted that there is an absence of guidelines for the implementation of WSN in dynamic or critical environments to allow natural disasters to be monitored without interruption, or for providing contextual information to adaptive systems. The implementation of WSN for monitoring this type of environment has more unique features than the implementations designed for other applications (e.g. static environments). In the context of WSNs when used for monitoring natural disasters, the reliability of the sensor nodes can be regarded as more important than the energy consumption when disasters occur. However, small and simple procedure codes are often found in the literature, as set out by Giannopoulos et al. (2009).

Our work aims to help fill in these gaps in the scientific literature and lay down guidelines that are based on experiences with the long-term and real deployment of a WSN for river flood monitoring in urban areas.

3. The WSN for our river monitoring system and the question of adaptation – long-term deployment

3.1. General reflections on the work carried out

Given the frequency of flash floods in Brazil during the rainy season and the serious loss of life and property this causes, we implemented and deployed a WSN-based river monitoring system. The WSN enables us to provide real-time measurement from the sensor nodes during critical periods (e.g. when there are torrential rainfall). This system is used by the emergency services and is crucial for issuing warnings or evacuating people form areas at risk.

The WSN architecture is called e-NOÉ and is designed to monitor urban rivers and issue warnings to the population (Hughes et al., 2011a). We deployed six sensor nodes along a creek in the city of São Carlos, in the state of São Paulo, Brazil (see Fig. 6 a for the deployment map). The nodes are equipped with analog sensors such as a river depth sensor, camera and rain gauge. They send data via a multi-hop ZigBee network (ZigBee Alliance, 2011) or 3G cellular network. One of the photos of these nodes is shown in Fig. 1.

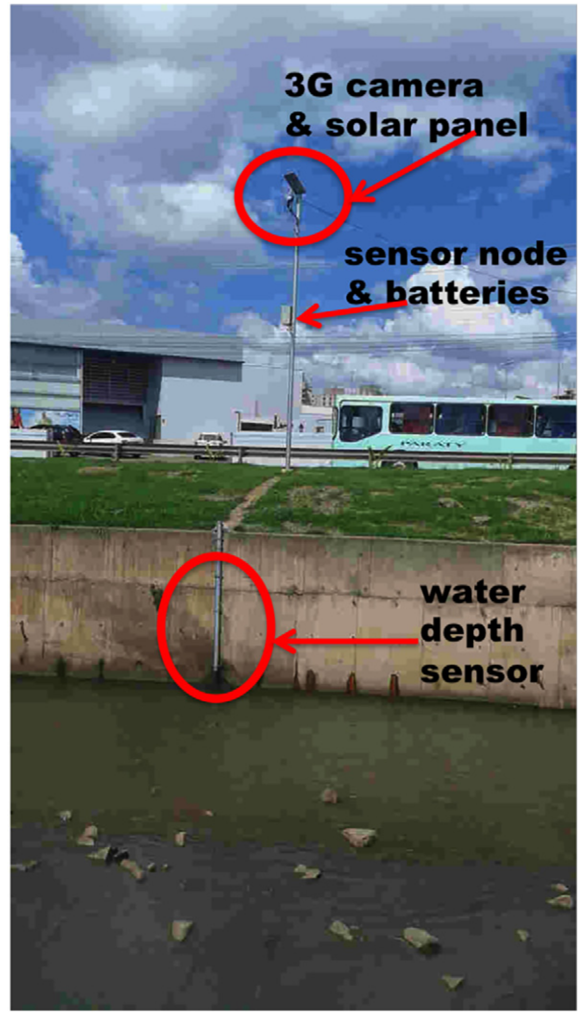

Fig. 1. IoT-based sensor node with camera and 3G modem deployed in the city of São Carlos, Brazil. This node is deployed with the RESS standard and located in an area with heavy traffic, therefore, we need to evacuate the area in case of floods.

When torrential rainfalls occur, the chances of failures in the WSN sensor nodes sharply increase as they are located along a river that is prone to floods. As a result, a part of the WSN will be unable to send information gathered from the environment because it will not have an alternative communication route to the sink node. To overcome this problem, Ueyama et al. (2014) recommended the use of Unmanned Aerial Vehicles (UAV) to provide the WSN with resilience when flying over the inoperative node and routing the messages received from other nodes in the network. As well as this, the UAV may be able to collect images in real time so that it can work together with the rescue teams. This is illustrated in Fig. 2.

This work has been conducted with the collaboration of the hydrologists from the Department of Hydrology and Sanitation at the University of São Paulo. Our research has been undertaken as a multidisciplinary study with a focus on both Computer Science and Hydrology.

3.2. Ongoing research on the WSN/IoT creek monitoring system

A number of key issues have been investigated in this research since we embarked on this project in 2010. These include the question of dynamic adaptation as well as how to devise a smart flood forecasting model. The study is geared to predicting floods rather than just detecting floods after they have already occurred.

The first results of research into flood forecasting are given by Furquim et al. (2016) that outlines the main results and the preliminary conclusions of his work. In this study, we aim at providing future estimates of the level of an urban river on the basis of the most recent data collected. In summary, the recommended model employs the Chaos Theory for the pre-processing of the historical time series and subsequently, a Multilayer Perceptron (MLP) to carry out the forecasting. The findings obtained in this study suggest that, in terms of the degree of precision required, good results can be achieved for the prediction of the level of the urban river up to 30 min in advance. Thus these results show that this model can be employed for predicting the level of the urban river in the near future and can assist in the decision-making after warnings have been issued to the public living in the region at risk so that they can be evacuated from the locality.

The use of participatory sensing (Burke et al., 2006) within the context of the river monitoring system is another area that is investigated in this research. Participatory sensing is a paradigm that involves requesting people to gather information from the use of mobile devices by collecting cooperative and collaborative data. In addition, we drew a human size image on the creek protection wall to make it easier to gauge the water level (see Fig. 3). It was hoped the drawing could help the volunteers by providing a visual aid to determine the water level. In Fig. 3, the white marks can be seen on the "person's" ankle, knee, waist, neck and above the head. These marks help volunteers to determine the level of the river, together with a smartphone that has an application with a copy of the illustration.

Creek image processing is also being employed for detecting floods since it can help us to detect and confirm the presence of floods instead of having to rely on depth sensors. As mentioned earlier, some of our nodes, particularly those deployed in critical areas, are equipped with cameras. Images are captured and sent to the base station periodically. However, new avenues of research have now

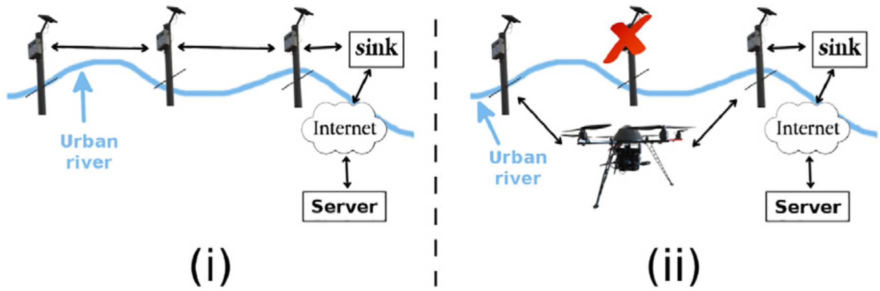

Fig. 2. Our UAV-based flood mitigation approach. (i) The WSN system sends collected data through wireless multihop communication to the sink node. (ii) If a node is unable to forward packets to its neighbor or a failure is detected, then a UAV is deployed either (a) to serve as a bridge between the nodes that became inoperative; or (b) to act as data mule between two communicating nodes.
Source: Text and image adapted from Ueyama et al. (2014) .

been opened up in the field of flood detection and prediction using creek images. We have already obtained the first results from this technique which are shown in Ortigossa, Dias, Ueyama, and Nonato (2015).[1]

Following the current trend, we have been exploiting the IoT (Internet of Things) technology (Santucci et al., 2009) in our flood monitoring system. This area has attracted our attention as a part of our plan to make our sensors available to any Internet user. By doing so, we can include, for example, any smartphone user who is driving, walking or at home/work. In our system, natural disaster warnings can be issued and this advanced notice can reduce the loss of life caused by floods/landslides in Brazil. We believe that the use of IoT can help us to integrate our system with the existing legacy systems that often rely on a TCP/IP stack protocol.

Hence, we combined an IoT and Sensor Web approach (Consortium, 2007) to ensure a higher degree of accessibility for those requiring the collected data. The people concerned include Internet users as well as hydrologists that do research in the field of urban creeks. To guarantee accessibility, we devised a portal, which can be seen at http://goo.gl/zqLEPP. Fig. 5 depicts our portal which combines IoT and Sensor Web systems. Our article in the Horita, de Albuquerque, Degrossi, Mendiondo, and Ueyama (2015) journal includes a fuller discussion of this line of research.

In this section, there will be a brief overview of the adaptation exploited in this research study. It should be stressed that we first devised our generic approach for runtime adaptation, which is outlined in Fig. 4. We then implemented several case studies in the form of component frameworks (CFs), which we discuss below. We also made use of an event-based component binding scheme called LooCI, which is described in Hughes et al. (2011b). The case study with LooCI was also conducted in the WSN-based urban river monitoring system.

4. A WSN as a reliable infrastructure for adaptable ubiquitous systems

4.1. RESS – Reliable Environment Sensing Standard

In this section, there is a description of our proposal which is a Reliable Environment Sensing Standard (RESS). This includes the guidelines that are essential to enable the algorithms to carry out the instructions in the sensor nodes so that a reliable execution can be ensured. Thus, our aim is to guarantee that the reliable infrastructure is implemented on the basis of WSN concepts for the monitoring of dynamic and/or critical urban environments. In this way, by allowing real-time information to be acquired even at the time of the disaster, the adaptive system can allow rational decisions to be made for the safety of the public.

RESS was developed by means of the concept of component frameworks (CFs) which was originally defined by Szyperski (2002) as " *collections of rules and interfaces that govern the interaction of a set of components 'plugged into' them* ". CF consists of rules and interfaces that are targeted at a particular application domain; in our case, a reliable WSN/IoT application scenario. Thus, a wide range of

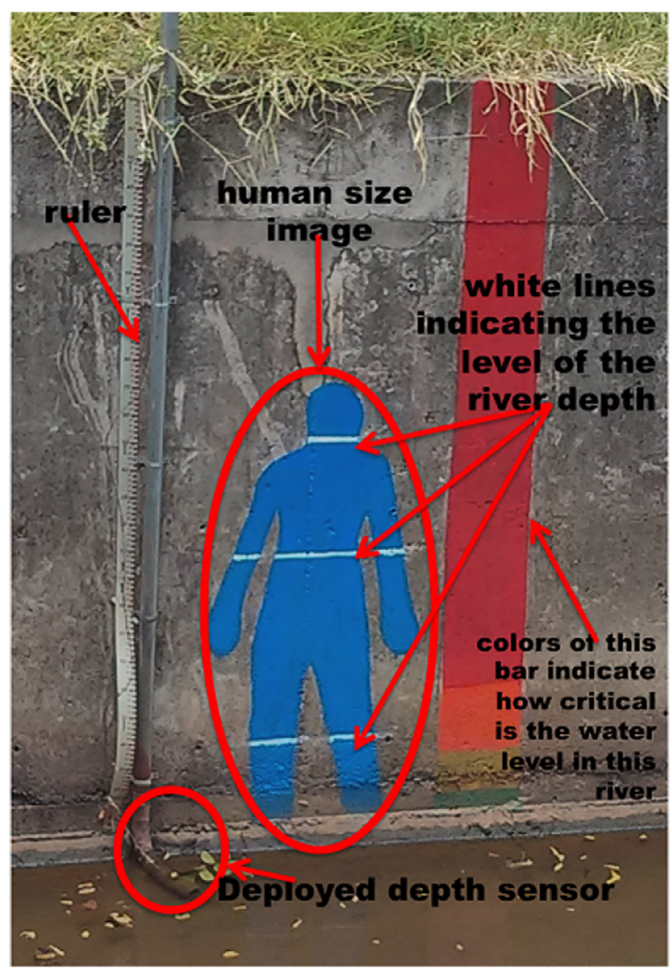

Fig. 3. Our approach to participatory sensing: the drawing on the wall of the real river illustrates a human with white marks that help volunteers to determine the level of the river height. The volunteer has a copy of the drawing on his/her smartphone. He/she taps on the white marks of the image displayed on his/her smartphone. Such action informs the depth level of the river and can help in determining whether there is a flood or not.

[1] Recipient of the best paper award.

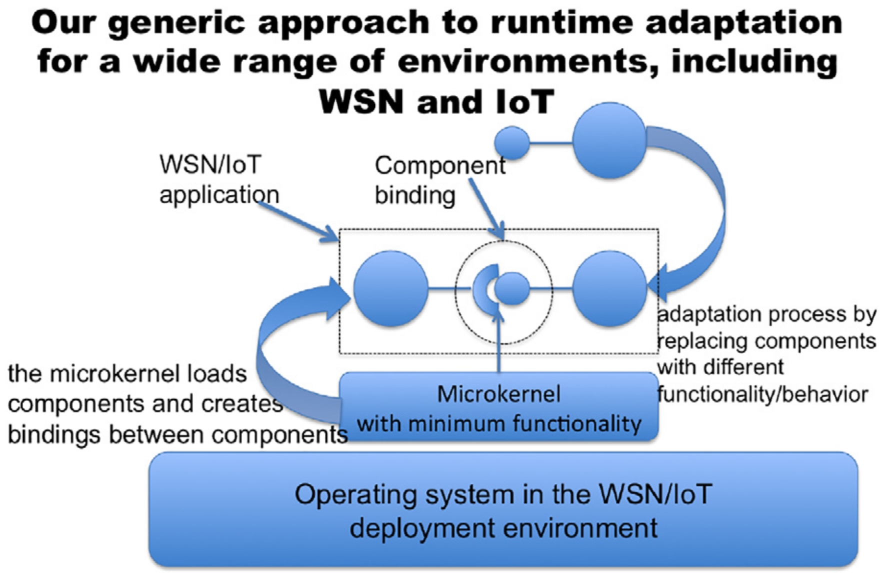

Fig. 4. In our generic adaptation approach, the microkernel is kept minimal so that it is deployable in a wide range of environments. It is only capable of deploying components and creating connections between them. Components can be replaced at runtime for tailoring the system to the target WSN/IoT need and application.

rules and interfaces are devised to provide a system that can operate within a critical scenario. A set of guidelines was drawn up to achieve this after five years' experience of deploying our WSN/IoT-based system. These guidelines are implemented in the form of CFs, as seen in Fig. 8.

The guidelines include technical and management factors to make sure there is a more reliable "critical system". The term "critical system" can be defined as one whose malfunction can cause death or serious injury to people. As a proof-of-concept, we embedded this new solution in a wireless sensor node and evaluated the behavior shown by both algorithms (non-RESS and RESS-based). From the obtained results, it can be stated that algorithms based on RESS increase the degree of availability of the monitoring system by approximately 48%, without any change in the hardware. Additionally, our proposal has enabled the sensor node to have a more stable behavioral pattern and greater fault tolerance.

We employed the ZigBee protocol, as it provides a stable set of algorithms for the network, transport and application layer. Even though is has been discredited by some researchers, it has a good performance when the energy consumption of the relay nodes is not the main concern. As our research focuses on the application, we can easily replace the ZigBee standard with a set of IETF-based protocols for high layer algorithms. We would like to point out, however, that even new standards such as Thread, rely on the IEEE 802.15.4 MAC layer,[2] which is also the basis of ZigBee networks.

4.2. Why the RESS model?

It is of crucial importance to have an infrastructure devoted to monitoring a particular region and, above all, that is reliable so that the adaptable systems can behave in a consistent way. A trustworthy system (in this case the WSN) must meet four essential requirements: (i) *Availability*: the likelihood of having a system that can operate correctly at any time and be available to carry out its functions; (ii) *Reliability*: the property of a system that ensures it can operate continually without any failure; (iii) *Safety*: the property that is geared towards a system that can be maintained and operated correctly; (iv) *Maintenance capacity*: the ability to repair a system that experiences a failure.

This study aims to examine procedures that can ensure a suitable behavior for the algorithms which will be executed by the sensor nodes in a dynamic and critical environment. This can be carried out by complying with the guidelines that we propose in the RESS model to develop a new algorithm (described in this study as a "proposed solution"). At the same time, the current algorithm (described in this study as the "existing solution") will be used as the basis for assessing the advances made through the use of our proposal. In both cases, the solutions will be implemented and embedded in a single WSN node so that the analyzed data represent the real behavior within the context outlined in this paper.

As previously explained, e-NOÉ is a WSN system installed in the town of São Carlos, state of São Paulo, Brazil, which is geared towards monitoring an urban river that occasionally floods during periods of torrential rainfall and causes a serious loss of life and property. The topology of this WSN consists of seven sensing nodes installed on the banks of the river, as illustrated in Fig. 6. The circles numbered from 1 to 6 are locations where there are nodes with pressure sensors that indicate the water level. The triangle shows the location where a sensor node is installed with a pluviometer to measure the intensity and amount of rainfall. Finally, the square refers to the base station that is responsible for receiving information from the sensor nodes and forwarding it to a database in a remote server. Communication between the nodes and the base station is based on the ZigBee protocol (Baronti et al., 2007). Although nodes 5 and 6 are equipped with ZigBee interfaces, they are out of range of the nearest nodes and, thus, use a 3G interface for communication. The main board is a Raspberry PI 2 model and it has connections with two other devices. The *first* is a pressure sensor with 15 PSI (103.42 kPa) Vented Gauge Male. The *second* device is an IP Camera with a 3G modem communication. The communication module is based on a Brazilian data service network. There is also a 900 MHz XBee wireless communication network and a UNIPOWER 12 V 7Ah battery set. The prototype has a KM20 20 W solar panel module with a 8A 12 V charge controller.

As can be noted, the region where node 6 is located is where most damage is caused during the period of torrential rains, since it receives a flow of water that originates from other regions which

[2] Thread group - https://threadgroup.org/About.

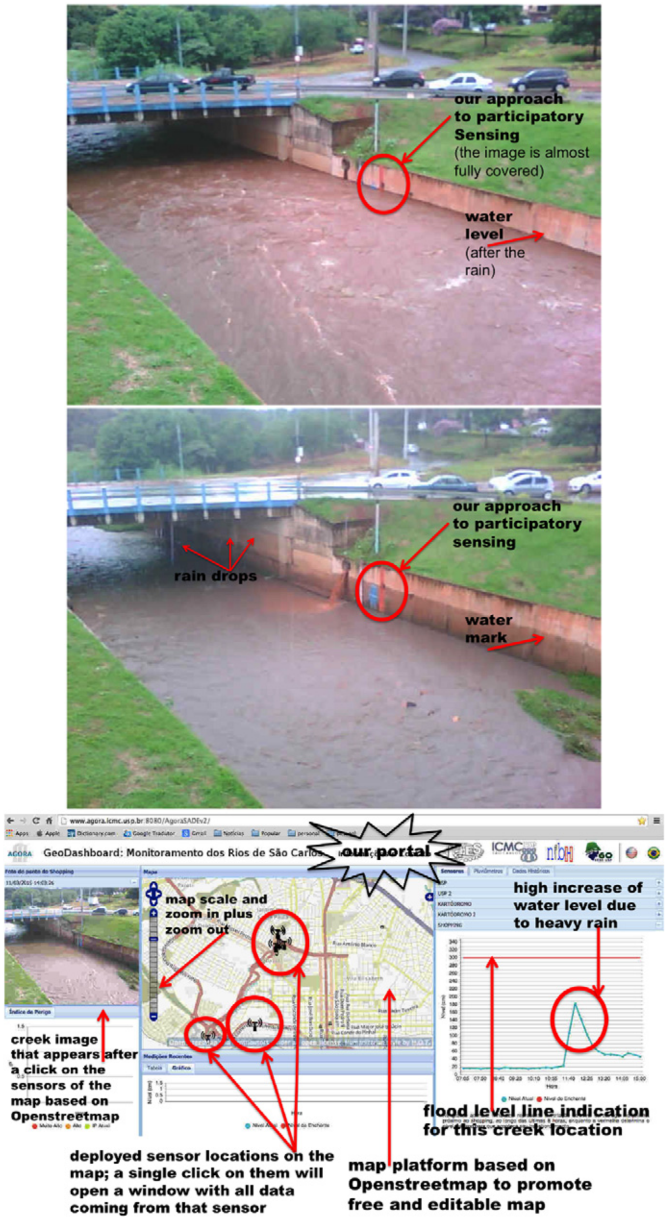

Fig. 5. Creek data and images being transmitted in real time by our IoT-based sensor node (node of Fig. 1). The monitoring took place under a torrential rain in March 3rd, 2015 - see also the difference of river level in the top two images (taken by our sensors), as well as the chart with a high increase in stream level, represented by the blue line of our portal.

are also being monitored (see Fig. 6). Fig. 7 shows the approximate boundary of the area of risk and includes a photo.[3]

4.3. The overall RESS architecture

In view of the fact that the existing literature focuses on the composition of complex adaptive ubiquitous computing systems within the context of smart cities, we decided to propose the RESS model. This refers to a **R**eliable **E**nvironment **S**ensing **S**tandard, which formalizes the guidelines for the creation of the algorithms that must be applied in the sensor nodes of a WSN that will be installed in a dynamic and critical region. Fig. 8 shows the overall scenario of our RESS architecture when it is deployed in a sensor node. The illustration is accompanied by the generic policies to ensure that the reliability is granted to this kind of critical application.

The overall sensor node architecture of RESS consists of the following features:

- *the deployment environment*: this consists of the hardware platform and the logic instructions (source code). The environment also has the RESS runtime that is instantiated in the sensor mode platform; RESS runtime is kept minimal to ensure that it is deployable in a large number of environments.
- *analog and network modules*: the analog sensors are included within the node to measure temperature and river depth and capture creek images (cameras are only deployed in nodes where flooding is very frequent). The communication can either take place by means of the ZigBee or 3G cellular network.

RESS refers to the guidelines to ensure that WSN-based smart city systems are reliable. It takes into account all the required policies to make sure that a reliable WSN-based smart city system is provided. In other words, RESS provides *what* a WSN-based smart city system should have to be reliable. The requirements can be grouped into the following items:

- *Managing existing resources*: the WSN-based smart city developer should take into account resource management concerns. These include allocating and de-allocating when they are no longer required. This is because these devices are generally resource-constrained.
- *Ensuring a correct code execution*: the systems should also consist of a mechanism to ensure that the smart city code is executed in accordance with the expected/planned behavior.
- *Implementing fault tolerance and recovery policies*: the WSN-based system should activate the deployment of a wide range of fault tolerance and recovery policies to ensure that they are triggered whenever demanded (e.g. be able to recover the prototype after a natural disaster has damaged it). The study of fault-tolerant WSN-based systems for a smart city is an ongoing work and has already been published (Beder, Ueyama, de Albuquerque, & Chaim, 2013).
- *Energy saving*: the system should be fitted with an energy conservation mechanism and ensure that a prolonged battery life is provided. Without this, the whole prototype will not be able to respond accordingly (i.e. the device should be kept in the sleep mode whenever possible).
- *Providing maintenance*: to have a reliable system, it is important to keep in mind that sensor node maintenance (i.e. a "check-up") is required from time to time even though the node appears to be working as expected (e.g. pressure sensor calibration is often needed after a period of time).

Note that RESS is developed as a CF, that can itself embody a number of other CFs. In this next section, there is a description of the policies that incorporate the RESS CF, as illustrated in Fig. 8:

- *Wireless communication policy*: this determines the technology adopted for the transmission of data and images (taken by the camera nodes). It is configured to use ZigBee or 3G depending on the needs of the target application and/or existing battery level.
- *Resource management code*: this provides the instructions for saving existing resources, which include, battery power, memory and processing cycles. In our scenario, the Internet of Things (IoT) policy saves less resources, while the WSN seeks to

[3] Photo courtesy of http://www.saocarlosagora.com.br press website.

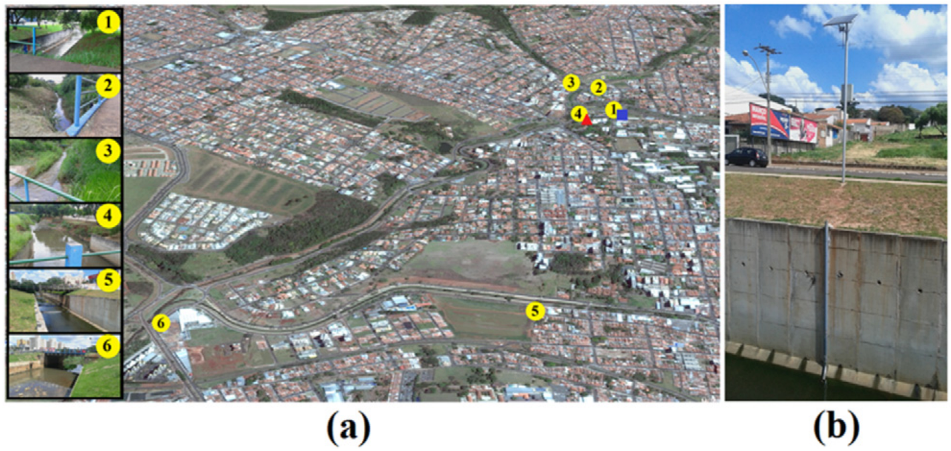

Fig. 6. Architecture and disposition of the sensor nodes of e-NOÉ along the bank of the urban river of the town of São Carlos, state of São Paulo, Brazil. In (a), the sites are highlighted where the sensor nodes have been installed. The numbered circles represent the nodes with pressure sensors. The triangle and the square refer to the node with the pluviometric sensor and the base station, respectively. In (b), there is a standard pattern of the numbered nodes. The post on the river bank supports a metal box in which the computing elements and the batteries (power supply) are stored and there is also a solar panel at the top, which is used for recharging the batteries. In this same image, it is also possible to see a metallic structure which reaches the urban river and stores the sensor pressure at its lower extremity.

save more resources, and is suitable for more resource-scarce environments.

- *Scheme for checking execution*: this lays down the rules for checking the outcome of the execution and takes measures accordingly. The "inspection" of the execution (particularly of the "critical" functions) is carried out to ensure it is run in a synchronous manner (i.e. this means a mechanism like ACK is required to respond in a way that confirms that it was successfully executed) or asynchronous fashion (i.e. no acknowledgment is required).
- *Fault tolerance*: this specifies what measures are required to manage the system when a fault is encountered. It might include mechanisms such as N-Version or Recovery Blocks to overcome faults; or it might incorporate policies for self-recovery, which can be instantiated when a fault is encountered.
- *Duty cycle instruction* indicates the amount of time that our system should be active. This will be adapted to the underlying sensor mode platform and can be configured from the *low power* to *high power* policy. In addition, the duty cycle can be adapted to the "critical" time that is imposed (e.g. increase the duty cycle during a period of torrential rain).

It should be emphasized that the above-mentioned items were derived from lessons learned after our WSN river monitoring system had been deployed in Brazil for five years. We came to the conclusion after years of discussion and making adjustments, that our research group was able to offer a more reliable prototype. After a serious drought in Brazil in 2014, this was followed by a rainy season that was the "heaviest" for the last ten years and it was found that our prototype functioned properly as a result of the adjustments made to meet new requirements. Hence, we believe that the above-mentioned items are of value, particularly those that include a WSN-based disaster management system in the context of a smart city.

4.4. Our previous and updated river monitoring sensing models (ensuring reliability)

There is a need for uninterrupted sensing so that the features of the ubiquitous computing system can be adapted to the context and improve public safety. Thus, it is necessary to include components that provide a stable execution and high availability in the firmware, as well as hardware and communication redundancy. The logic of the existing solution is shown in Algorithm 1, and it was initially embedded in node 6. No verification or checking is required if the operation is carried out successfully. Moreover, during the transmission of the collected data (on the level of the river and image), the 3G modem may need high-voltage electric power. If the feeding source does not supply this voltage, the node is restarted, by interrupting

Fig. 7. Region monitored by node 6. In (a), there is a boundary of the approximate area of risk and in (b) there is a photo published by the local press of the flood that occurred on March 14, 2011.

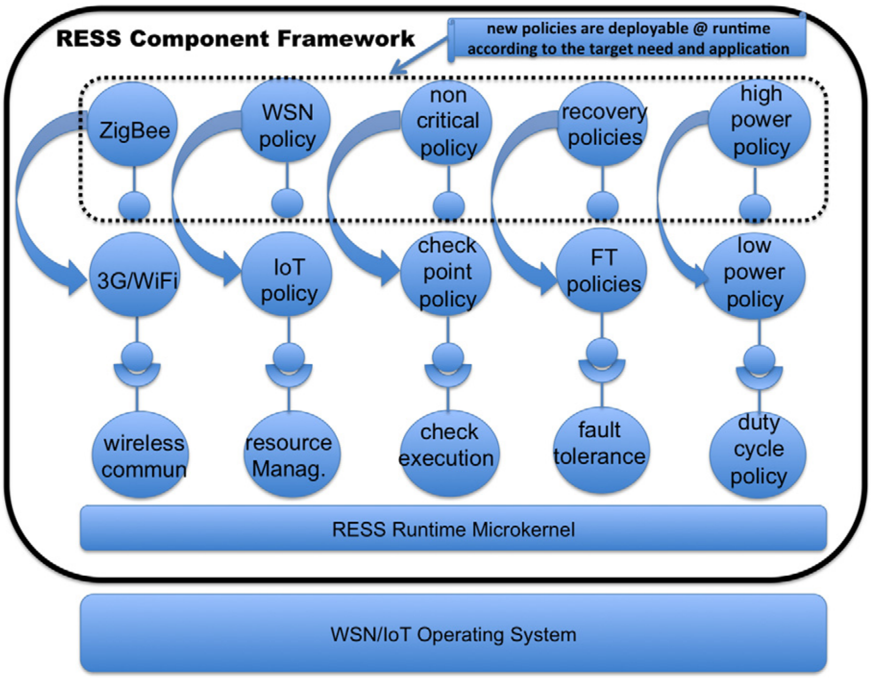

Fig. 8. The RESS component framework architecture devised after five years of real deployment in Brazil; and the adaptation scenarios.

the transmission and not releasing the allocated space. However, a situation of this kind can have an adverse effect on the inner storage space of node 6 when all the space available is allocated and this can prevent the data from being collected and sent. As well as this, the sensor node is kept in execution mode the whole time and this may cause failures because of the stress on the system. This results in a fragile behavior and susceptibilty to failures, which means that node 6 and its data are not available 100% of the time.

Algorithm 1. Algorithm of existing solution to the sensing carried by node 6.

```
begin
    while TRUE do
        if Last Sensing >= 5 minutes then
            Turn on Modem3G();
            Set Modem3G();
            Search date and time on the Network();
            read ← 0;
            for i=0; i<=5; i←(i+1) do
                read ← read+read_Sensoring();
            end
            level ← read/5;
            Transmits(level);
            CaptureStoresPhoto();
            Transmit(Photo);
            Delete(Photo);
            Turn off Modem3G();
            Count(5 minutes);
        end
    end
end
```

This problem can be addressed and lead to a reliable WSN by employing the Reliable Environment Sensing Standard (called RESS) which formalizes a number of guidelines with a view to controlling the development of robust solutions geared towards sensor nodes installed in critical and dynamic environments. On the basis of our proposal, it is possible to design solutions (and subsequently implement them as firmware variants) and these lead to a significant increase in the availability of sensor nodes. In addition, our proposal seeks to assist by ensuring that the solutions can lead to a greater availability without altering any of the hardware used for the implementation of the WSN.

In view of this, the required guidelines for the development of robust algorithms for dynamic and critical environments can be listed as follows:

- *Always check the current available resources* - this is because processing cycles for checking current available resources often do not require as much energy as using wireless communications. When carrying out this check, the system also has to update its available resources and be able to respond promptly whenever required so that it can carry out a task on behalf of another node (e.g. encapsulate and forward another node's data to the sink node).
- *Carry out garbage collection periodically* - as our sensor nodes keep running for a long period of time (particularly during the rainy seasons); inevitably garbage is accumulated during this period of time. This means, the system must collect garbage from time to time to avoid clogging the nodes. On a number of occasions, our nodes just "froze" because the memory was full of garbage.
- *Verify the correct code for execution* - whenever possible, the system should verify the code that ensures correct execution. This can be done by checking the return codes generated by the executed functions.
- *Save as much energy as possible* - the system should be kept in "sleep" mode whenever (and as long as) possible. This is still the "best" measure to save energy in a WSN node. The process of saving energy also includes avoiding unnecessary processing.
- *Employ fault recovery methods* - the system should be able to recover from a fault whenever it is detected. For example, if a code encounters an endless loop, originated from a programming error; the system should be able to reboot the entire node

to recover from the errors. This line of research is a continuation of the study undertaken in our previous work (Beder et al., 2013).

The formalized guidelines in our proposal are designed to attain the following goals: (i) to prevent errors caused by the lack of internal resources; (ii) to determine the degree of success of the operations that are carried out; (iii) to save energy; (iv) self-recovery from failures without human intervention; (v) to provide stability in the behavior of the sensor node; (vi) to avoid stress in the device; and (vii) to ensure a high degree of availability of the information collected by the sensor node. In this way, we were able to ensure that the algorithms created on the basis of RESS, fulfilled three of the four requirements that are necessary for a trustworthy system (availability, reliability and maintenance capacity).

On the basis of our proposal, the existing solution was modified. Algorithm 2 shows the changes that were made and the logic behind the proposed solution.

Algorithm 2. Algorithm of proposed solution to the sensing carried by node 6.

```
begin
    while TRUE do
        if Last Sensing >= 5 minutes then
            return ← QueryFilesStored();
            if return == TRUE then
                files[] ← ListFiles();
                foreach files[] do
                    DeleteFiles(files[]);
                end
            end
            Turn on Modem3G();
            Set Modem3G();
            Search date and time on the Network();
            read ← 0 for i=0; i<=5; i←(i+1) do
                read ← read+read_Sensoring();
            end
            level ← read/5;
            Transmits(level);
            CaptureStoresPhoto();
            Transmit(Photo);
            Delete(Photo);
            Turn off Modem3G();
            Sleep(5 minutes);
        end
    end
end
```

It should be noted that Algorithm 2 checks if there is any file stored in the internal memory before the instructions for the collection and transmission of data are carried out. If there is one, a list is created with the name of all the files for later deletion. This operation ensures that there will be enough space for temporary storage of the collected data until they are transmitted. In addition, at the end of the execution cycle, node 6 goes into "sleep" mode for 5 min. This procedure makes it possible to save energy, avoid unnecessary processing and maintain the device without stress.

The recommended modifications were implemented in the firmware and embedded into node 6. The employed methodology to measure the behavior with both solutions and to analyze the results obtained, are described in the next section.

5. Performance evaluation

5.1. Evaluation methodology

A new version of firmware was generated to assess our proposal, which was based on the solution and embedded into node 6 (see Fig. 6). Node 6 was chosen as a proof-of-concept since it is in a node located at the edge of the network and would not impair the communication of the other sensor nodes if there was a problem of unavailability. However, it should be noted that node 6 is more affected by rainfall owing to the fact that the current of the river brings all the water to this location.

Node 6 supplies data about the water level from a submerged pressure sensor. This data is sent directly to the server which stores the information in a database through a 3G connection, at intervals of 5 min. As a result, this assessment will address the question of the availability of node 6 to supply data, in view of its importance in predicting floods (Furquim et al., 2016). The readiness of node 6 to supply images of the urban river will not be addressed in this paper, as it is part of another ongoing project.

The availability rate of node 6 is calculated on the basis of information recorded in the server during the period January 2014–February 2015.[4] The records represent the information available about environmental conditions. If the information is unavailable, it is assumed that some failure has occurred and caused the unavailability of node 6 as a source of information about the environment. Since our solution was installed in November/2014, the data for this month were not used in the analysis. This is because node 6 was being upgraded to enable the proposed solution to be put into effect; several tests were conducted with the aim of determining what adjustments were needed for the implementation.

The records carried out at the database have the date and time at which the urban river was "sensored". On the basis of these registers, it was possible to ascertain the interval between the monitoring and these could be used to calculate the total number of records that should exist in ideal conditions (with 100% availability).

Eq. (1) was used to calculate the daily availability of node 6. In this equation, λ is the time interval in minutes which must exist between each register (in each solution, this interval is defined as 5 min); h and m are two constants equal to 24 and 60, respectively. It is worth pointing out that with the information described up to this time, it was possible to calculate the total number of expected registers in ideal conditions (in this case, 288 registers). Finally, x is the number of registers that exist in the database with regard to the sensing that took place on a particular day.

$$\beta d = \frac{x}{\frac{h*m}{\lambda}} * 100 \qquad (1)$$

However, when account is taken of the sensing period used for the analysis in this paper, it can be stated that the number of registers is very large. In view of this, they were assigned to the month in which they were carried out so as to make it easier to analyze the results and arrange them clearly. Thus the analysis conducted in this paper is based on the monthly availability rate of node 6.

5.2. Discussion of results

In calculating the monthly availability, Eq. (1) was altered to Eq. (2) where X is the total number of registers that have references to a particular month and d is equal to the number of days in each month. Thus, Eq. (2) was used to calculate the monthly availability of all the months of the period in question, except for the month of November/2014 (i.e. the updated procedure for the proposed solution).

$$\beta m = \frac{X}{\frac{d*h*m}{\lambda}} * 100 \qquad (2)$$

[4] Data analyzed in this paper are available in http://goo.gl/pSu5nH.

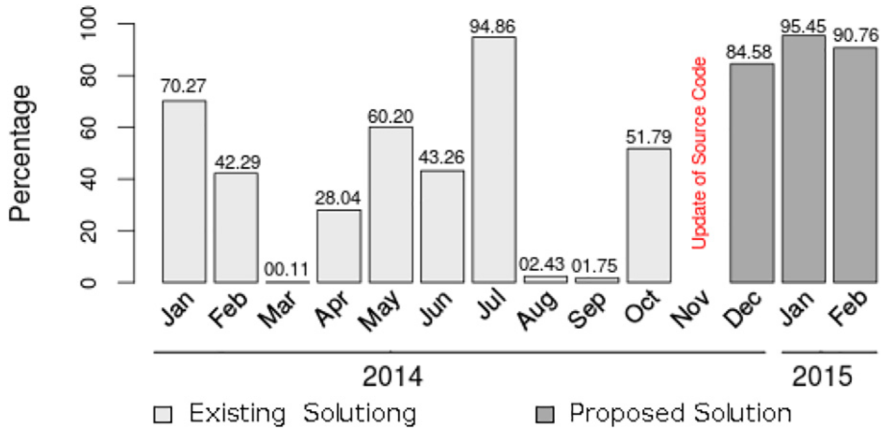

Fig. 9. Graphic representation of the availability of node 6 with regard to the time line on which the registers were analyzed. The solution being used by node 6 is shown through the colouring of the bars (light grey for the existing solution and dark grey for the proposed solution).

Fig. 9 shows the monthly availability rates (in percentages) of node 6 by comparing finding the existing solution and the proposed solution. It should be noted that the results are arranged along a timeline from Jan/2014 to Feb/2015.

In the results displayed, it can be noted that there is an imbalance between the samples. The reason for this is that it was decided to use a larger sample for the behavior at node 6 with the existing solution, so as not to affect the analysis since in the last three months (Aug-Oct/2014) the availability of the sensor node was very low. In turn, the sample of the behavior of node 6 with the proposed solution is lower because there are no more registers available on a monthly basis until the collection time for the analysis. In this way, we sought to maintain a true equilibrium between both samples.

As can be noted in Fig. 9, the proposed solution increased the availability of node 6 significantly, without any alteration to the hardware. Although there were still periods of unavailability at node 6, it resumed its normal monitoring behavior without the need for technical maintenance (which was necessary with the existing solution).

It can be seen that the proposed solution tends to allow a greater degree of stability in the monthly availability rates, as can be seen in Fig. 10. This is apparent from the dispersal of the values of the availability rates which are represented by boxplots. In addition, the sample median of the data for each solution shows a value of ≈42.77% for the existing solution and ≈90.76% for the proposed solution, which shows a significant gain in the monthly availability rate when the proposed solution is used.

This interpretation can also be reached by examining the behavior of the sensor node displayed on March 27, 2015 when a serious flood took place at node 6. The sensor node at that date functioned properly without interruption during the entire period of torrential rain. Collected data and images are available at http://goo.gl/vbBmsR. Fig. 11 shows the information provided by node 6 before, during and after the disaster. Additionally, Fig. 12 shows the photograph obtained by node 6 during the disaster. However, we wish to include a log system at node 6 to assist in determining and quantifying the problem that must be overcome.

A wide range of statistical methods have been employed to validate these results and they are shown in Table 1. At first we used the Shapiro Wilk method to check the adequacy to normality of the availability rates obtained from the evaluated solutions (existing and proposed) and, depending on the results, to define the use of parametric or non-parametric methods. Both the solutions had a p-value greater than 0.05, with 0.503 for the existing solution and 0.849 for the proposed solution. In this way, the hypothesis of adequacy to normality was approved with a confidence level of 95%. Hence, the parametric tests can be highly recommended for any subsequent analysis.

In the next analysis we used the *t* -test, with the aim of determining whether there is a significant statistical difference between the availability rates shown for the solutions. By employing this method, it was possible to ascertain that the p-value between the samples is 0.001, which suggests a significant statistical difference between the proposed solution and the existing solution. As a result, it can be statistically determined that the use of the RESS guidelines made it possible to achieve higher monthly availability rates, as well as a more stable behavior.

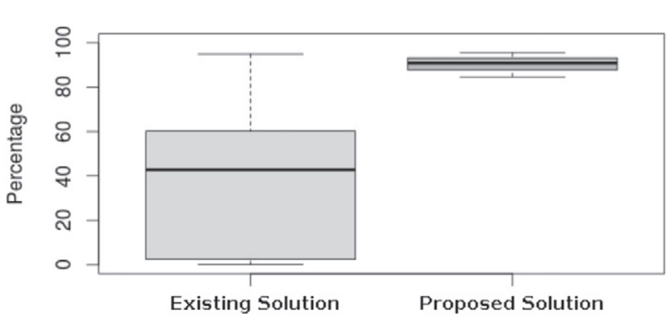

Fig. 10. Boxplot of the monthly availability rates of node 6 with the use of both solutions.

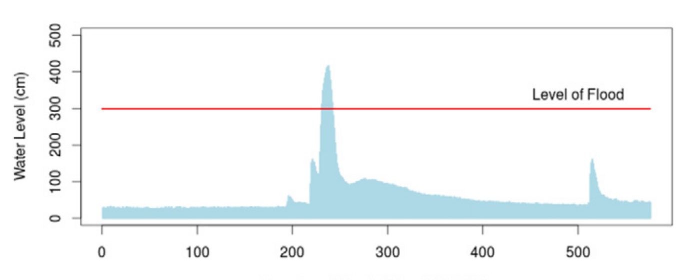

Fig. 11. Graphic depicting the sensored data by node 6 which gives an overall creek height before, during and after the flood that took place on March 27, 2015 (running the proposed solution). The red line indicates the flood level.

Fig. 12. Pictures taken by the e-NOÉ architecture of a dynamically critical region. With these images, it is observed that a particular region can be safe or dangerous, depending on weather conditions.

6. Conclusion and suggestions for future studies

This paper has made a proposal of an adaptable and reliable WSN-based system for critical applications, such as a flood monitoring system. In putting forward this: (*i*) we discussed the key issues and requirements for ensuring that a reliable WSN system is obtained. Second, (*ii*) we examined how these adaptable and critical systems should be developed so that they can be tailored to the targeted system application and hardware platform.

The key issues and requirements are given through the RESS standard that we devised after deploying a WSN-based river monitoring system for five years. With regard to the adaptable approach, we recommended using the generic system that we have been adopting for a wide range of software such as mobile phones and WSN nodes. The adoption of our approach ensures that the constructed system can be tailored to a wide range of target applications.

We also examined our long-term WSN-based river monitoring deployment system and provided an extensive survey of all the work carried out to date in this line of research. This included exploiting participatory sensing, UAV-based flood mitigation and Sensor Web-based portals in our deployment. We also outlined how the use of LooCI and OpenCom middlewares could make WSN environment programming easier. These environments are often recognized as being hard to program. With regard to the RESS results, we should point out that RESS leads with an average monthly availability rate of ≈90.76%, which represents an increase of ≈48% compared with an algorithm that does not follow its guidelines.

Finally, it should be emphasized that despite the good results obtained from our proposal, there still remain periods of unavailability. For this reason, we intend to investigate a number of areas in future studies by: (i) employing a methodology to determine the main failures in WSNs in dynamic and critical environments; (ii) exploring techniques that can be applied to measure the level of safety in this kind of environment by taking account of events in the past and/or near future; and (iii) examining the question of resource redundancy for WSNs in chaotic environments.

Table 1
Statistical analysis for the validation of the given results.

Statistical analysis			
Shapiro Wilk		*t*-Test	
Solution	p-Value	Solution	p-Value
Existing	0.503	Existing × proposal	0.001
Proposal	0.849		

Acknowledgments

The authors would like to thank Geraldo P. R. Filho for his comments and advice, which were of great value when undertaking this research project. The authors would like to thank CNPq, CAPES and FAPESP for funding the bulk of this research project. Bruno S. Faiçal would like to thank FAPESP (process ID 2013/18859-8) for its financial support. Jó Ueyama would like to thank The Office of Naval Research Global (ONRG) for funding a part of his research project.

References

Araujo, G. B., Duarte-Figueiredo, F. d. L., Tostes, A. I., & Loureiro, A. A. (2014). A protocol for identification and minimization of traffic congestion in vehicular networks. *2014 Brazilian symposium on computer networks and distributed systems.* (pp. 103–112). IEEE.

Baggio, A. (2005). Wireless sensor networks in precision agriculture. *ACM workshop on real-world wireless sensor networks (REALWSN 2005).*. Stockholm, Sweden.

Baronti, P., Pillai, P., Chook, V. W., Chessa, S., Gotta, A., & Hu, Y. F. (2007). Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards. *Computer Communications*, *30*, 1655–1695.

Beder, D. M., Ueyama, J., de Albuquerque, J. P., & Chaim, M. L. (2013). FlexFT: A generic framework for developing fault-tolerant applications in the Sensor Web. *International Journal of Distributed Sensor Networks*, *9*, 385892.

Burke, J., Estrin, D., Hansen, M., Parker, A., Ramanathan, N., Reddy, S., & Srivastava, M. (2006). Participatory sensing. *Proceedings of the world sensor web workshop, ACM SENSYS. Boulder, Colorado*.

Castillo-Effer, M., Quintela, D. H., Moreno, W., Jordan, R., & Westhoff, W. (2004). Wireless sensor networks for flash-flood alerting. *Devices, circuits and systems, 2004. Proceedings of the fifth IEEE international Caracas conference on. Vol. 1.* (pp. 142–146). IEEE.

Consortium, (2007). OGC sensor web enablement: Overview and high. *Technical report*.

Furquim, G., Pessin, G., Faiçal, B. S., Mendiondo, E. M., & Ueyama, J. (2016). Improving the accuracy of a flood forecasting model by means of machine learning and chaos theory. *Neural computing and applications*, *27*(5), 1129–1141.

Giannopoulos, N., Goumopoulos, C., & Kameas, A. (2009). Design guidelines for building a wireless sensor network for environmental monitoring. *2009 13th panhellenic conference on informatics.* (pp. 148–152). IEEE.

Gomez, L., Laube, A., & Sorniotti, A. (2008). Design guidelines for integration of wireless sensor networks within enterprise systems. *ACM digital library, editor, in the proceedings of the international conference on on mobile wireless middleware, operating systems, and applications*.

Horita, F. E., de Albuquerque, J. P., Degrossi, L. C., Mendiondo, E. M., & Ueyama, J. (2015). Development of a spatial decision support system for flood risk management in Brazil that combines volunteered geographic information with wireless sensor networks. *Computers & Geosciences*, *80*, 84–94.

Hughes, D., Ueyama, J., Mendiondo, E., Matthys, N., Horré, W., Michiels, S., … …Guan, S.-U. (2011a). A middleware platform to support river monitoring using wireless sensor networks. *Journal of the Brazilian Computer Society*, *17*(2), 85–102.

Hughes, D., Ueyama, J., Mendiondo, E., Matthys, N., Horré, W., Michiels, S., … …Guan, S.-U. (2011b). A middleware platform to support river monitoring using wireless sensor networks. *Journal of the brazilian computer society, springer*. *Vol. 17*. (pp. 85–102). Springer.

Kim, Y., Jabro, J. D., & Evans, R. G. (2011). Wireless lysimeters for real-time online soil water monitoring. *Irrigation Science*, *29*, 423–430.

Komninos, N. (2011). Intelligent cities: Variable geometries of spatial intelligence. *Intelligent Buildings International*, *3*, 172–188.

Mampentzidou, I., Karapistoli, E., & Economides, A. A. (2012). Basic guidelines for deploying Wireless Sensor Networks in agriculture. *2012 IV international congress on ultra modern telecommunications and control systems*. (pp. 864–869). IEEE.

Neto, J. T., Guidoni, D. L., & Villas, L. (2014). A new solution to perform automatic meter reading using unmanned aerial vehicle. *2014 IEEE 13th international symposium on network computing and applications*. (pp. 171–174). IEEE.

Ortigossa, E. S., Dias, F., Ueyama, J., & Nonato, L. G. (2015). Using digital image processing to estimate the depth of urban streams. *Workshop of undergraduate works in conjunction with conference on graphics, patterns and images (SIBGRAPI), Salvador, Ba brazil*.

Rashid, B., & Rehmani, M. H. (2016). Applications of wireless sensor networks for urban areas: A survey. *Journal of Network and Computer Applications*, *60*, 192–219. http://linkinghub.elsevier.com/retrieve/pii/S1084804515002702. http://dx.doi.org/10.1016/j.jnca.2015.09.008.

Rehmani, M., & Pathan, A. (2016). *Emerging communication technologies based on Wireless Sensor Networks: Current research and future applications*, CRC Press. https://books.google.com.br/books?id=TsH1CwAAQBAJ.

Santucci, G. (2009). From internet of data to internet of things. *International conference on future trends of the internet*. (pp. 1–19).

Seal, V., Raha, A., Maity, S., Mitra, S. K., Mukherjee, A., & Naskar, M. K. (2012). A simple flood forecasting scheme using wireless sensor networks. *International Journal of Adhoc, Sensor & Ubiquitous Computing (IJASUC)*, 45–60.

Sunkpho, J., & Ootamakorn, C. (2011). Real-time flood monitoring and warning system. *Sonklanakarin Journal of Science and Technology*, *33*, 227.

Szyperski, C. (2002). *Component software: Beyond object-oriented programming*. Addison-Wesley.

Ueyama, J., Freitas, H., Faical, B. S., Filho, G. P., Fini, P., Pessin, G., Gomes, P. H., & Villas, L. A. (2014). Exploiting the use of unmanned aerial vehicles to provide resilience in wireless sensor networks. *IEEE Communications Magazine*, *52*, 81–87.

United Nations (2014). World urbanization prospects: The 2014 revision. *United Nations, Department of Economic and Social Affairs (DESA), Population Division (2014). World urbanization prospects: The 2014 revision, CD-ROM edition.*.

Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. *Computer networks*, *52*, 2292–2330.

ZigBee Alliance (2011). ZigBee. http://www.zigbee.org.