*Journal of Ubiquitous Systems & Pervasive Networks*
*Volume 5, No. 2 (2014) pp. 21-28*

# Privacy Management Solution in Ubiquitous Environments Using Percontrol

**Valderi R. Q. Leithardt[1], David Nunes[2], Anubis G. M. Rossetto[1], Carlos O. Rolim[1], Cláudio F. R. Geyer[1], Jorge Sá Silva[2]**

[1] *Federal University of Rio Grande do Sul - Institute of Informatics – GPPD - Porto Alegre, RS - Brazil*

[2] *University of Coimbra Department of Informatics Engineering Pólo II, Pinhal de Marrocos, Coimbra - Portugal*

Abstract

The identification of users in pervasive and ubiquitous environments can be performed using various devices and techniques, among which the use of wireless sensor networks (WSN), radio frequency identifiers (RFID), Smartphones, Bluetooth, among others. However, it is necessary to implement management techniques and control privacy of these heterogeneous environments. Thus, we performed a study using artificial neural networks in order to perform the processing of information identified only generating an output of information to Percontrol.

***Keywords:*** *Pervasive Computing, Routing, Wireless Sensor Networks, Artificial Neural Networks.*

## 1. Introduction

In 1991, Mark Weiser [1], claimed that the most profound technologies would become so integrated onto our daily lives that it would be as if they just "disappeared", since using them would become second-nature. With this idea in mind, Weiser proposed a new form of computation, pervasive computation.

By "disappearing" it is meant that the device becomes imperceptible; in other words, the user no longer needs to worry about the technological resource in use. From this perspective, pervasive computing frees users from worrying about secondary tasks (using the device's interface correctly) and allows them to focus on their primary task (use the device as a tool to perform the desired action).

The development of microelectronics and wireless communication has greatly benefited the development and usage of Wireless Sensor Networks (WSNs), namely in a pervasive / ubiquitous environmental context.

WSNs are usually composed by a large number of wireless nodes spread through an area containing the events / phenomenon to be monitored [2]. A sensor node is a device with reduced size meant to be cheap and easily deployed in risky or difficult to access locations. Their small size severely limits their hardware capabilities; sensor devices usually present many

limitations when it comes to battery, processing power, storage and communication interfaces [3]. These restrictions affect the amount of sensor nodes present in a given network, as well as the network's performance. This leads to a need to develop communication protocols that can handle the sensors' inherent limitations while being adequate for the application and associated routing mechanism. Being a sub-class of ad-hoc networks, routing is a task handled by the sensors themselves, and not managed by a specific routing device. By cooperating amongst themselves the sensors can carry data throughout the network in an energetically efficient way, through wireless radios, light pulses or ultra-sound. Sensorial data is transmitted throughout the network in "hops"; a sensor transmits its data to the next sensor in the network closer to the data's final recipient, trusting that the transmission will occur smoothly, without interruptions. This transmission mechanism is useful to distribute the transmission's energy costs throughout the whole network, avoiding the need to use high potency signals [2].

This makes the use of WSNs in pervasive/ubiquitous environments a more complex challenge, since these environments are populated with many other types of devices and communication mechanisms [1]. All this diversity in devices and communications is managed by the pervasive / ubiquitous application; an example of such an application is Percontrol [4], a pervasive system for controlling user attendance through mobile devices and wireless networks. This work's main contribution is an analysis of the viability of using Artificial Neural Networks (ANNs) as a base technique to handle user profiles and manage devices on a pervasive environment that also uses WSNs to capture environmental data such as luminosity, temperature and movement. The results from this viability analysis are novel and serve as initiative to further explore this concept in future research works in the area of pervasive computation.

* Corresponding author. E-mail: vrqleithardt@inf.ufrgs.br

DOI: 10.5383/JUSPN.05.02.004

The article is organized as follows: section II presents the Percontrol system and introduces several important concepts regarding ANNs and WSNs. Section III describes some of the most widely known concepts of routing in WSNs. Section IV gives an overview of the state-of-the-art in the area while section V presents the application scenario. Section VI contains information on the performed tests and shows some preliminary results. To finalize, section VII gives a conclusion and alludes to future work.

## 2. Percontrol

Percontrol [4], [5] is a system that manages and keeps track of user attendance in an automatic way; in other words, the system detects the entrance and exit of users within an environment, be it academic or entrepreneurial. Percontrol also bolsters a user discovery and localization service, within the local environment, that is based on Bluetooth, WIFI and RFID. On an academic environment, the system controls student attendance without the teachers intervention; on an enterprise environment, it controls the entrance and exits times of the employees without the need for their intervention. By resorting to certain aspects of pervasive computing, the system developed in [5] can guarantee reliability and transparency in the detection and localization of users, resulting not only in minimal impact on both the classroom activities or the workers' service, but also avoiding potential forgetfulness of entrance and exit controls. The system identifies users through their mobile devices, and controls the environment based on their previously-defined profiles. The user / device presence on a certain location generates an event that is sent to a WebService listener application. This listener application firstly identifies the user within the environment and then from configures the locale in accordance to the user's profile stored in the system's database. Environmental data such as temperature and luminosity are acquired through SunSpot wireless sensors [6] in conjunction with Arduino development kits [7]. In the work described in [5], an earlier version of this system was already being developed, but the management of different user profiles and devices on a single environment was yet to be supported. In order to support this feature, an artificial intelligence module was necessary which is now supported through the use of neural networks.

Thanks to its newfound intelligence, the system is now capable of adapting the environment to fit the demands of several users, simultaneously. The ANNs receive data from each user, perform pattern recognition and, by resorting to previous event data, make a decision to configure the environment in a way that is satisfying to most users.

Solving problems with ANNs is an attractive approach. According to [16], the way these problems are internally represented in the network and the natural parallelism associated with ANNs' architecture create the possibility of a superior performance when compared with traditional machine-learning methods.

In ANNs, the usual procedure for problem-solving begins with a "learning" phase, where a set of examples are presented to the network from which are extracted the necessary characteristics to correctly represent the information provided [8]. These characteristics are then used to generate answers to the problem in question.

An artificial neural network is composed by several simple processing units. These units are connected through communication channels that have a certain "weight" associated to them. Each processing unit performs computational operations upon the data that is fed to it through its connections. The intelligent behavior that results from an artificial neural network comes from the interactions between the network's processing units.

The most important property of ANNs is the ability to learn from their environment (that is, the input data) and improve their own performance. The entire "knowledge" contained within the network is stored in the "synapses" between the networks artificial neurons; in other words, the networks behavior is mostly conditioned by the weights associated to the connections between individual neurons. Hence, the development of a computational solution that not only infers a system's current state but also predicts its future condition seems to be the solution for the environment management problem previously mentioned. Considering the above characteristics, a neural network is a prediction mechanism that fits Percontrol's needs, since it can use raw contextual data, learn from it and predict the configuration values that will better fit a future situation. The solution proposed in [9] was used as basis to adapt a neural network to the current application scenario. This effort demanded an analysis of the characteristics of WSNs in order to find the best configuration and techniques to associate ANNs parameters to Percontrol WSNs.

The main components of sensor networks are the phenomenon, the sensor and the observer. The phenomenon is the event of interest to an observer, and it is monitored and processed by a WSN. Several different phenomena can be observed concurrently by a WSN. Monitoring is performed under certain criteria that determine the performance requirements for the sensor nodes to be used; this way, the acquisition and dissemination of information throughout the WSN is made in an effectively.

The sensor device is used to collect the data generated by the phenomenon and transmit it to the next node through wireless communication. Nodes respond to changes in the environment where they are currently operation, such as climatic changes, etc. The characteristics and capabilities of the sensor depend on the application and each node's sensibility may differ, depending on the distance to or the exposure time to the event being monitored. The observer is the element that receives the information collected by the WSN; depending on the application's needs, there may be more than one observer at any given time [10].

WSNs can be classified according to their configuration, sensing mechanisms, communication medium and processing type [2]. The type of routing is one of the most important research topics for reducing energy consumption and increase performance in wireless sensing applications.

## 3. Routing

Due to the peculiarities of their utilization (such as their limited battery life), communication between sensor nodes, deployed in an external environment, is subjected to many obstacles that may impede its effectiveness. Most applications that use WSNs communication are directed towards configurations with stationary characteristics, which differ from traditional Ad-Hoc networks, Routing can have its focus on addresses or data, the latter being a more recent approach whose main advantage is the fact that data aggregation manages to reduce the amount of packets exchanged in the network, and thus, reduce the overall energy consumption of the system [11].

Another approach to routing in WSNs is related to the TinyOS operating system, developed by the University of Berkeley. TinyOS does not use a complete addressing scheme like traditional IP-based Ad-hoc networks [11], [12]. Traditional Ad-hoc networks implement a routing protocol that constructs a "Minimum Spanning Tree" from an initial sink or access point [12]. The limited resources of a WSN stress the need to use simplified addressing schemes that reduce the size of network packets and the amount of RAM memory needed to store routing information.

Routing in WSNs can be categorized as: based on MAC (Medium Access Control), plane routing, hierarchical routing and geographical routing. MAC-based routing is based on conventional a network that applies techniques to minimize some of the most critical problems in WSNs. The main routing protocols that use medium access control according to [13] are: S-MAC (Sensor-MAC), ARC (Adaptive Rate Control), T-MAC (Time-out-MAC), B-MAC (Backoff-MAC), DE-MAC (Distributed Energy Aware MAC), TRAMA (Traffic Adaptive Multiple Access).

In plane routing, all nodes are considered as equals from a functional point-of-view; routing activity is shared fairly amongst all nodes in the network, without a hierarchical organization. Some plane routing protocols are presented in [2], [5] and [10]. SPIN is a protocol for WSNs that uses information about the amount of energy available in each sensor to manage routing operations. Negotiation protocols are used to disseminate information from a single node to all nodes in the network. Other routing protocols are Directed Difusion, SAR (Sequential Assignment Routing), Adaptative Local Routing Cooperative Signal Processing, Multi, PROC (Proactive Routing with Coordination), SID (Source-Initiated Dissemination) and STORM/AD. The Mica Motes from University of Berkeley use the "TinyOS Becoming", a routing protocol directed towards limited hardware to be used in stationary homogeneous networks, where sensory data is periodically sent to a base station. This protocol assumes sensor nodes to have very limited processing and communication capabilities and presents a very compact implementation.

In hierarchical routing there are two main classes of nodes: sources nodes and cluster heads. Source nodes simply collect and send data to their respective cluster head, which may then perform fusion / aggregation tasks before sending collections of data to the gateway [9]. All nodes within a class are considered equal from a functional perspective. Examples of these routing protocols are: LEACH (Low Energy Adaptive Clustering Hierarchy), TEEN (Threshold Sensitive Energy Efficient Sensor Network), APTEEN (Adaptative TEEN), SHARP Hybrid Adaptive Routing Protocol and PEGASIS (Power-Efficient Gathering in Sensor Information Systems). PEGASIS is a protocol for WSNs that is based on the concept of "currents". Each node exchanges information only with its closest neighbors, forming a current between several nodes, whereas a single of these nodes is chosen to pass the collected information to the gateway.

Geographical routing uses geographical information to route data; this information usually includes neighbor node location [12]. Localization data can be referenced from a Global Position System (GPS) or from a local reference, valid for all the nodes in a network or just for a subset of neighbor nodes. The main geographical algorithms are LEACH–C, ICA (Inter Cluster Routing Algorithm), Geographic Routing without Location Information and GeoMote (Geographic Multicast for Network Sensors). In GeoMote there are three types of sensor nodes: GeoHosts (which produce data), GeoRouters (which foward data collected by the GeoHosts) and the GeoGateways (which act as data entry and exit points) [4]. There is also GEAR (Geographical and Energy Aware Routing) and GPSR (Greedy Perimeter Stateless Routing). Unlike the previous protocols, GPSR allows the addressing of a single node [9]. It uses two algorithms for data routing: when a single node identifies a neighbor that is closer to the final destination, the node sends data to that neighbor. If there is no closest neighbor, the data packet should be sent to a more distant node, in order to avoid areas of poor node-coverage. In these situations, the protocol builds a planar graph to identify the best neighbor to forward the data to [14].

In this section, a resume of some important routing protocols used in WSNs was given. An exhaustive comparative simulation of routing protocols for WSNs is presented in [14], which was used as theoretical basis to formulate the tests and obtain the preliminary results that serve as validation for the work here presented.

## 4. State-of-the-art

As previously mentioned, this article's proposal refers to an implementation of ANNs in a pervasive control system for pervasive/ubiquitous environments in conjunction with WSNs (composed of SunSpot wireless sensors, for testing purposes). The system deals with concurrent data and adapts the environment's configuration to match its users' profile, according to the devices used to perform login onto the environment.

In the literature [15] we can find several academic works that focus on the detection of devices in many application areas within the domain of pervasive systems and wireless sensor networks, as introduced by Mark Weiser. The work in [15] shows a system that implements several integrated systems that allow the monitoring of patients in their own home. However, this work does not address the integration of different forms of communication within the same environment nor does it present mechanisms for handling user profiles. In Carvalho [16] an extensive proposal that focus on elderly users in ubiquitous environments is presented; the Internet is shown as a form of interaction but this work does not specify which languages, devices and forms of communication should be used. RIBEIRO [17] proposed a project that consisted on the creation of an artificial intelligence system capable of leading a robot through certain routes. The system was tested using a robot-vehicle that possessed freedom of movements (horizontal locomotion). The environment was scanned using sensors whose data was fed to the artificial intelligence, where an algorithm evaluated the situation and acted on the locomotion engines, making the robot avoid possible obstacles and follow the desired route.

Table 1. Comparison between related research

| Support and Control / WORKS | Scope | Devices Transmission | Language | Public | Routing Protocol |
|---|---|---|---|---|---|
| ARAUJO | ubiquitous computing | Not defined | Not defined | Not defined | Not defined |
| CARVALHO | ubiquitous computing | Environmental sensors, measuring devices, electronic devices, Internet | Not defined | Elderly people who require constant monitoring of their health | Not defined |
| RIBEIRO | Neural Networks | Sensor end-of-course and sonar | C/C++/C#, Java | Not defined | Not defined |
| FONSECA | Neural Networks, pervasive systems | RFID | Delphi | Laboratory of the University | Not defined |
| PERCONTROLE | Neural networks, systems, pervasive / ubiquitous | WiFi / RFID / Bluetooth Wireless Sensor Networks. | .Net | Educational and business environments | Zigbee |

Despite being an innovative proposal, a target audience and usage scenario are not defined, hence not adapting to the proposed model. Fonseca [18] defines a solution that uses neural networks and RFID, partially achieve the desired results. Despite the idea of uniting neural networks and RFID in pervasive environments being valid and commendable, one of the weak points of this proposal was the language used, while the existence of other devices and user profile management tasks are not considered.

Percontrol presents all the necessary features for pervasive / ubiquitous environment usage, considering different types of devices, and forms of communication. This work presents a significant improvement over previously presented versions in [4] and more recently in [5], while addressing all the primary objectives previously set. Table 1 presents the main characteristics and functionalities of the works researched in the available literature.

## 5. Application Scenario

The application scenario demonstrates the potential pervasive / ubiquitous computation has in improving efficiency in the workplace. It also attempts to illustrate different possible perspectives one can have on a single pervasive scenario. Initial versions of Percontrol did not anticipate the use of WSNs or ANNs [6], such versions only intended to automatize teacher's student-attendance tasks in classrooms. Teaching is a teacher's main occupation, while keeping track of student attendance is a necessary, but also rather inconvenient secondary task that is prone to errors, interrupts the class, wastes teaching time and may have to be performed multiple times in order to prevent students from leaving. Hence, Percontrol focuses on performing detection of people that enter the classroom environment, hence performing student tracking automatically. This work proposes an extension of the work developed in [6] and [7], increasing the pervasive functionalities available in this user-tracking system, with the objective of increasing control over environmental conditions through the user's mobile devices. Using SunSpot wireless sensors [14] and Arduino kits [7] Percontrol can sense and manage the temperature and luminosity of an environment; by using ANNs, the system can also attempt to adjust the values of these environmental properties to fit the individual preferences and the number of users in the environment, turning it into an intelligent location. The sequence diagram in Figure 1 shows the primary interactions between all parts of the system, as well as the messages that are exchanged since a user is detected until the environment adapts to his preferences. When the application detects the entrance of a device in the environment, a webservice that manages the associations between users and devices is accessed. The device is identified through its BDA (Bluetooth Device Adress), WiFi or RFID. The application maintains a module called BlueID which holds a list of all devices that were ever detected. Each time the application verifies the devices currently present in the environment, it performs a comparison with the previously stored list; newly detected devices generate an "entry" event while missing devices are associated with an "exit" event.

When accessed, the Webservice returns to the application the person's username, associated device resources and personal preferences, through the HTTP protocol and an XML format message. The application also communicates with the SunSpot sensors to fetch the room temperature, luminosity, humidity or other environmental data that may be used at a later time. The following format was used to communicate with the sensor: messageID : sensorTypenn. Both the "messageID" and the "sensorType" are numerical values. The messageID field is used to associate the sensors response with the respective BlueID request, an important step since communication is asynchronous.

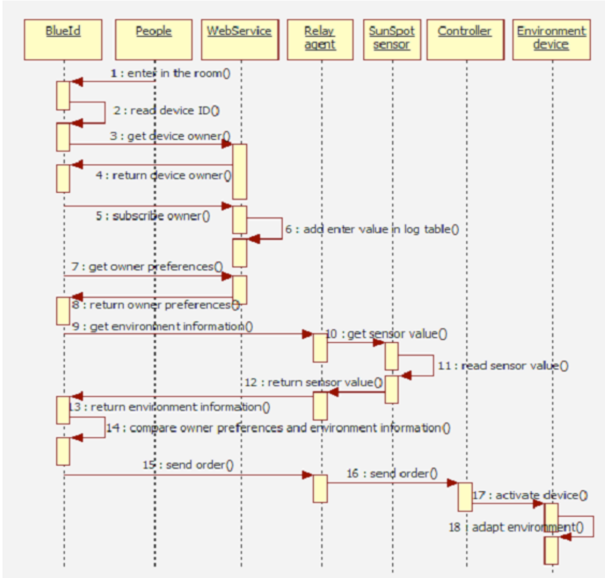


**Fig. 1. Sequence Diagram**

The sensorType represents the type of data being sent (luminosity, humidity, etc); the "\n" character is used to mark the end of a message, while the ":" character is used as a data separator. The current environmental state is compared with the user's environmental preferences in order to decide what changes need to made. After a decision is reached, the environment sends commands to the actuator controllers (connected through USB to an operating computer) in order to change the environmental characteristics (by turning the AC unit on and change the room temperature, for example). The extension of Percontrol functionalities translated into a more complex architecture, as it is shown in Figure 2.

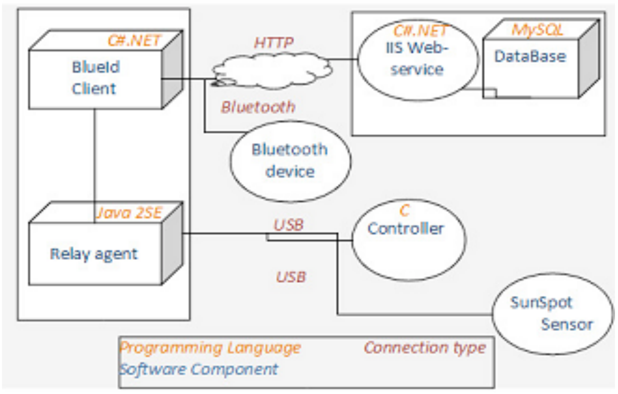


**Fig. 2. System's Architecture**

Initially, the prototype application and its respective transmitters were tested on a Windows operating system, an environment that benefited from the use of SunSpot sensors [6] and Arduino hardware [7]. There were many other advantages that led us to choose the Arduino boards, namely the embedded input/output ports, low cost and strong modularity (using the appropriate "shield" extensions). The main idea behind the use of Arduinos was to test their viability for middleware development in pervasive environments, not excluding the possibility of having these boards completely replace the various individual sensors for an integrated, single-board solution connected to a computer. Figure 3 illustrates the functioning of our application.

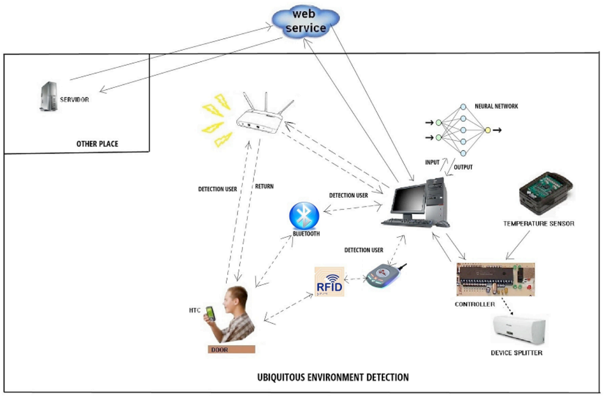



**Fig. 3. Functioning of Percontrol**

From Figure 3, we can see that there will be a need to implement a central controller that will allow the exchange of profiles in the environment, according to the user's preferences. The main challenge here is related to the number of parameters generated by the application while using WSNs and ANNs, which increase as the amount of sensor nodes increases. This results in larger energetic and resource demands, as well as in increased complexity for the neural network processing, which may compromise the ANN's response time. An efficient monitoring of the network's performance is necessary to guarantee a good quality of service. An example of this can be observer in the amount of time necessary to obtain information regarding a monitored environment; if it takes too long to obtain environmental information, this information may lose its value from an application perspective. Management of performance may provide means for the application to define proper quality metrics. These may be influenced by node density, exposition time, amount of dissipated energy and other factors, as specified in the description of the study presented in section 3. A mechanism that evaluates the level of importance of information is necessary for the management of quality of service. For instance, a sensor detecting a temperature of 20±C during spring is a normal occurrence. However, a the measurement of 50±C under the same circumstances is an abnormal event, which turns this measurement into a relevant situation that requires an extra dose of attention; it could implicate an artificial intelligence mechanism that compares the abnormal value with values measured by other sensors to see if the information is reliable and to determine the proper course of action. Information that is of great importance to the normal function of the system should imply a greater effort for proper delivery. That is, the energy consumed in communications should vary depending on the importance of the data. Another relevant management aspect concerns the installation of ad-hoc networks in unknown areas, where the behavior of wireless communications can be highly unpredictable with high error-rates and considerable delays, which may compromise the value of the information to the application. Performance management usually includes the following sets of functions: quality assurance, performance monitoring, performance control, performance analysis [2]. The QoS management process begins with the detection of performance degradation and ends when the source of the problem is removed. In between, the process has many intermediary stages of situation analysis [19]. Initially used only 2 tests kits sunspots containing two wireless sensor nodes communicating with each base station connected to the computer via USB, with that comes the need to conduct a comparative study of routing protocols for use in different environments and with numbers over wireless sensors. To this end, several techniques exist to treat this problem and allied service discovery, one of the most important, is used by the Protocol Service Location Protocol (SLP) [14], which basically consists of maintaining a directory that contains the services available to those who is offering them. However, it is necessary to study thoroughly the operation of routing protocols in order to verify the protocol that best fits the pervasive system of control required and its peculiarities, it is not the objective of this work the study of routing protocols. Therefore, we conducted some tests to validate the survey and obtained results that demonstrate the feasibility of work and consequently their implementation and use in the application percontrol, contributing to the improvement of the system and data so that other researchers can use.

## 6. Tests and obtained results

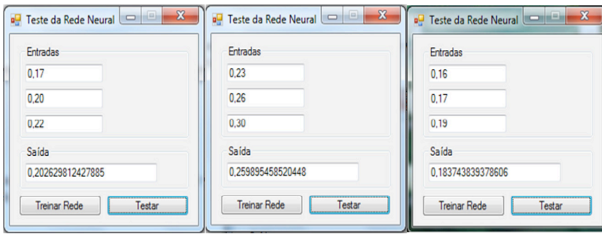
One issue with having multiple users on the same system is the problem of concurrent data; for example, the configuration of an air-conditioning unit may be influenced by every user that registers in its environment, since each person might have their own preferred temperature, whereas the temperature itself is constant throughout the whole environment. In order to bypass this problem, the decision-making process for selecting the best "average" temperature must take into account the individual preferences from all users within the environment. A widely used solution [20], [9], [21] that has shown great results is the use of AI, in particular ANNs [8]. A neural network bases itself on real data that has occurred in previous moments and that has been stored within the system for posterior access and use.

The main objective of this work is not the choice of proper protocols or AI tools, but the creation of novel helper mechanisms for our pervasive application. Our choice for an AI mechanism resided on neural networks, while the routing mechanisms were TCP-IP and ZigBee. These choices were supported by other published works in the areas of routing protocols [14], artificial intelligence [21], and comparison and use of neural networks [22] and [23]. The neural network loads the entire history of a device being handled within the environment, and uses its historical data as "training", in order to identify decision patterns that were assumed in a recent past. Considering our air-conditioning example, these patterns include the temperature that each person wants for a certain environment and what temperature was actually used when all users were taken into consideration. This type of analysis is crucial for the network's decision-making process. Figure 4 presents part of the source code used to define the desired and assumed temperatures. These values are fixed for testing, but in a real scenario they are fetched from a database or an archive.

```
    // Inicializa com
    //   1 randon
    //   3 perception neurons (entradas)
    //   6 hidden layer neurons
    //   1 output neuron (saída)
    net.Initialize(1, 3, 6, 1);

    // Taxa de aprendizagem
    net.LearningRate = 3;

    // Iterações
    iterations = 10000;

    // Treina a rede
    net.Train(input, output, TrainingType.BackPropogation, iterations);
}
```

**Fig. 4. Part of the neural network's source code**

The code shown in Figure 4 is used to train the neural network and we can see that following the training phase, the next step is to test the network to determine if it is apt to solve the problem of finding the ideal temperature from past event data. In order to perform the testing, a graphical interface was developed. The interface receives the values for current data and returns the ideal temperature estimation, as shown in Figure 5.




```
private void Testa()
{
    // Define os parâmetros para testar a rede
    net.PerceptionLayer[0].Output = Convert.ToDouble(txtE1.Text);
    net.PerceptionLayer[1].Output = Convert.ToDouble(txtE2.Text);
    net.PerceptionLayer[2].Output = Convert.ToDouble(txtE3.Text);

    // Pulsa a rede para obter o retorno
    net.Pulse();

    txtS.Text = net.OutputLayer[0].Output.ToString();
}
```

**Fig. 5. Training the ANN**

After the neural network's training, we could identify the network's response time after a user enters the environment, as shown in Figure 6.

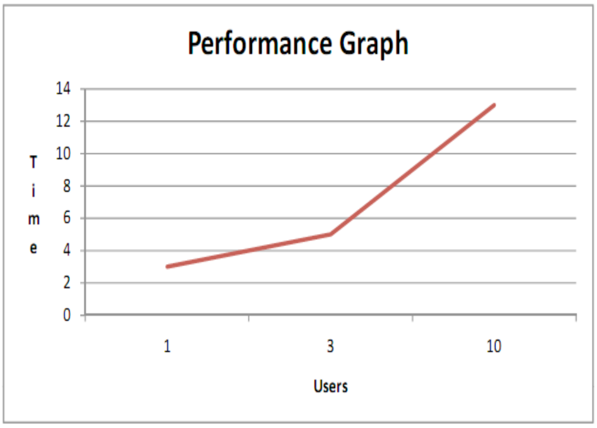



**Fig. 6. Neural network's performance and response time**

Figure 6 presents the response times of the ANNs for the cases with 1, 3 and 10 distinct users identified by the pervasive environment, where the X axis represents the number of users and the Y axis represents the elapsed time. For a single user, the ANN took 3 seconds to process the information contained in the user's profile, returning an average temperature with a value equal to the one defined by the user (since it is just a single person). For three users, the neural network took 5 seconds to respond, whereas for ten users it took 13 seconds. In Figure 7 a screen is shown, containing information on the users identified by the system, as well as on the devices associated with them. To perform the identification of different environments we used an Arduino Duemilanove [7]. The Arduino Duemilanove ("2009") is a microcontroller board based on the ATmega328, that possesses 14 digital input/output pins (6 of which can be used as PWN outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. The board contains the necessary assets to support the microcontroller and using it is as simple as connecting it to a computer with a USB

cable or powering it with an AC-to-DC adapter or battery. With this board it was possible to detect devices via Bluetooth, WIFI and, after being integrated with an appropriate card reader, through RFID. The reader fetched the RFID a card's serial number that can be cross matched with the user's registration on the database. To this end, a RFID card reader model YHY502CTG was used in conjunction with the Arduino board.

After obtaining the necessary application data and performing the necessary adjustments we validated our system using a didactic MultiPIC development Kit which possesses its own internal programmer. In other words, the MultiPIC microcontroller can be programmed

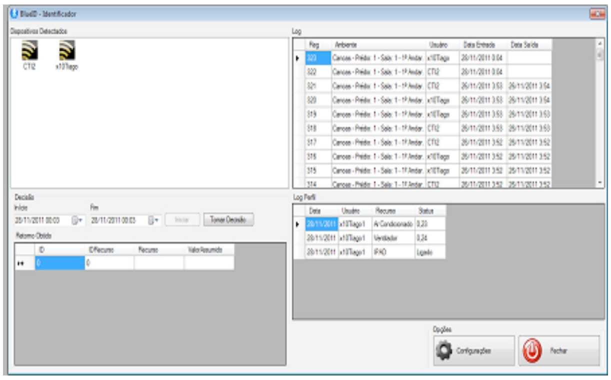


**Fig. 7. User identification screen**

directly on the board during the software's development phase; there is no need to remove the controller and perform its programming on separate equipment. The didactic development Kit was connected a stepper motor that simulated a ventilator. The stepper motor can be put in action with different speeds; initially we defined 3 different speeds that corresponded to 3 different profiles. The software used to simulate a ventilator with 3 speeds that controlled the board and stepper motor was developed in C language using the development software from Microchip MTLab, and transferred to the microcontroller with the IC-Prog 1.06C, the software used to compile the source code onto the MultiPIC kit's processor. On the performed tests the ANNs computed the average temperature from the user profiles and used current environmental information from the SunSpot sensors to correctly manage the ventilation system. From these tests we conclude that Percontrol managed the pervasive environment in a satisfactory manner and that the primary objectives of this research were met, although there is still much room for improvements.

## 7. Conclusion and future works

As suggested in previous sections, the choice of a routing protocol should be based on the restrictions inherent to the observed phenomenon which define the monitoring environment and the minimum requirements for the sensing hardware. The main purpose of the work and results herein presented are to be used as basis for future research work that improves the area of pervasive / ubiquitous computation, namely in the use of ANNs with low response times in environments with thousands of users.

The performed tests confirm the viability of device detection with WIFI, Bluetooth and RFID, an improvement over previous Percontrol versions. Nevertheless, there is still some latency in registering new devices on the system, which may be reduced by further adjustment in the parameters sent to the ANNs. Still, the presented work represents a significant contribution since it covers different areas and technologies within pervasive computation. On top of several parameters and definitions still waiting implementation, future work needs to define a model and implement a mechanism for privacy control, for both users, their devices as well as for the pervasive environment itself.

## Acknowledgments

This work is associated with the UbiArch project - Ubiquitous Architecture for Context Management and Application Development Federal University of the Rio Grande do Sul. We would like to thank the Conselho Nacional de Desenvolvimento Cientifico e Técnologico CNPq for its support. We would also like to thank Sun Microsystems for donating the SunSpot development kits, which allowed us to perform the necessary tests for our study. We kindly thank the students Sebastien Skorupski, Thiago Tomasel, Marceli Stein and Jeison Nunes da Silva which contributed towards the completion of this work.

## References

[1] M. Weiser, “The computer for the twenty-first century.” Scientific American, vol. 265, no. 3, pp. 94–104, Setembro 1991. http://dx.doi.org/10.1038/scientificamerican0991-94

[2] Loureiro, A. A. F. et al, Redes de Sensores Sem Fio. Disponível em: http://www.sensornet.dcc.ufmg.br/pdf/179LoureiroNogueira_Ruiz_Mini_Nakamura_Figueiredo.pdf. Last Accessed. Jun 2012.

[3] Luz, D.L., “Roteamento em Redes de Sensores”, IME - USP, São Paulo, 2004.

[4] Ayres, C; LEITHARDT, V. R. Q; Everson Figueiró; Hunder Correa; Luis Henrique Leal Ries. Controle Acadêmico Pervasivo. In: XXI Salão de Iniciação Científica (UFRGS), 2009, Porto Alegre. Controle Acadêmico Pervasivo. Porto Alegre: Editora da UFRGS, 2009.

[5] Leithardt, V. R. Q.; Rolim, C.; Rosseto, A.; Geyer, C.; Dantas, M. A. R.; Silva, J. Sa; Nunes, D. "Percontrol: A pervasive system for educational environments". In ICNC 2012, Hawai US.

[6] http://www.sunspotworld.com/ - Last Accessed Out 2012.

[7] http://www.arduinobrasil.com/ - Last Accessed Jul 2012.

[8] Carlos O. Rolim; Anubis Rossetto; Valderi R. Q. Leithardt; Cláudio F. R. Geyer. "Análise de uma Rede Neural Híbrida como base para um Mecanismo de Predição de Situação". in Congresso da Sociedade Brasileira da Computação (CSBC 2012) Simpósio de COmputação Pervasiva Ubíqua. Curitiba, Brasil 2012.

[9] Silva, R. N., Leithardt, V .R. Q., Silva, J. Sá, Geyer, C.F.R; Joel, J. R., Boavida, F. “A Comparison of Approaches to Node and Service Discovery in 6lowPAN Wireless Sensor Networks”. Tenerife, The Canary Islands, Spain; The 5th ACM International Symposium on QoS and Security for Wireless and Mobile Networks.

[10] Augusto, C. H. P., Carvalho, C. B., Silva, M. W. R., and de Rezende, J. F. - "REUSE: A Combined Routing and Link Scheduling Mechanism for Wireless Mesh Networks", to appear in Computer Communications (COMCOM), 2011. ISSN: 0140-3664. English, A4 size, 10 p.

[11] Cabrini, F.H., Kofuji S.T., “Roteamento em Redes de Sensores Sem Fim”, Laboratório de Sistemas Integráveis-USP, São Paulo, http://www.pad.lsi.usp.br/humanlab/trabalhos/workshop_roteamento.Pdf

[12] Lima, dos Santos, L. Um protocolo para descoberta e seleção de recursos grades móveis Ad Hoc, Tese de Doutorado, PUC RIO 2007, Pág. 27 e 28.

[13] Anubis G.M. Rossetto; Carlos Oberdan Rolim; Valderi R.Q. Leithardt; M.A.R. Dantas; Cláudio F.R. Geyer: "An adaptive fault tolerance approach to enhance the execution of applications on multi-cluster grid configurations from mobile grid interfaces in wireless networks" in Int. J. of High Performance Systems Architecture 2011 - Vol. 3, No.4 pp. 202-215; DOI:10.1504/IJHPSA.2011.045505. http://dx.doi.org/10.1504/IJHPSA.2011.045505

[14] LEITHARDT, V. R. Q; GEYER, C. F. R; Tavares. Estudo Comparativo entre Protocolos de Roteamento Utilizados em Redes de Sensores sem Fios. In: Congresso Trinacional da Fronteira, 2009, Foz do Iguaçu. IV Congresso da Acadêmia Trinacional da Fronteira. Foz do Iguaçu: PTI Itaipu Binacional.

[15] ARAUJO, Regina Borges.(2003) Computação Ubíqua: Princípios, Tecnologias e Desafios. Departamento de Computação – Universidade Federal de S. Carlos (UFSCar). Disponível em http://im.ufba.br/pub/MAT570FG/Livros eArtigos/045_AraujoRB.pdf . Last Accessed set 2012.

[16] CARVALHO, Sergio Teixeira. COPETTI, Alessandro, LOQUES, Orlando Gomes. Sistema de computação ubíqua na assistência domiciliar à saúde. Journal of Health Informatics (JHI), vol. 3, no. 2, p. 51-57, 2011.

[17] RIBEIRO, Dionisio R. de Souza, MACHADO, Kássio L. da Silva, PEREREIRA, Levi C. Azevedo. Rede Neural Perceptron embarcada em robótica. Instituto de Estudos Superiores da Amazônia. Belém, 2008.

[18] FONSECA, Lucas Oliveira. Sistema de localização RFID de usuários visando à racionalização de energia em ambientes inteligentes. Universidade de Brasília. Brasília, Feb. 2011.

[19] Pereira, V.; Silva, J.S.; Granjal, J.; Silva, R.; Monteiro, E.; Qiang Pan; , "A Taxonomy of Wireless Sensor Networks with QoS," New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on , vol., no., pp.1-4, 7-10 Feb. 2011.

[20] BRAGA, Antonio de Padua; LUDERMIR, Teresa Bernarda: "Redes Neurais Artificiais: Teoria e aplicações", (2007), Rio de Janeiro: LTC.

[21] LUDWIG JR., Oswaldo; COSTA, Eduard Montgomery Meira. Redes Neurais: fundamentos e aplicações com programas em C. 1 ed. Rio de Janeiro: Ciência Moderna Ltda, 2007.

[22] SILVA, R; SILVA, S. J; GEYER, C. F. R; LEITHARDT, V. R. Q; BOAVIDA, F. Use of GPS and 6LowPAN in Mobile Multi-Sink Wireless Sensor Networks - Issues and Feasibility.. In: Proceedings of the 8th International Information and Telecommunication Technologies Symposium (I2TS), 2009, Florianópolis - Brasil.

[23] Rolim, C. O; Fernando Schubert; Rossetto, Anubis G.M; LEITHARDT, V. R. Q; GEYER, C. F. R; Westphall, Carlos. “Comparison of a Multi output Adaptative Neuro-Fuzzy Inference System (MANFIS) and Multi Layer Perceptron (MLP) in Cloud Computing”. In: XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2012, Ouro Preto. X Workshop em Clouds e Aplicações - WCGA. Ouro Preto. SBC, 2012.