Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

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

Review

Real-time data management on wireless sensor network: A survey

Q1 Ousmane Diallo [a], Joel J.P.C. Rodrigues [b,*], Mbaye Sene [c]

[a] Department of Mathematics and Informatics, Instituto de Telecomunicações, University of Beira Interior, UCAD, Senegal, Portugal
[b] Instituto de Telecomunicações, University of Beira Interior, Portugal
[c] Department of Mathematics and Informatics, UCAD, Senegal

A R T I C L E   I N F O

Article history:
Received 22 July 2011
Accepted 5 December 2011

Keywords:
Wireless Sensor Networks
Real-time database management
Data collection
Survey

A B S T R A C T

In the recent past, search in sensor systems focused on node hardware constraints and very limited energy resources. But nowadays, that new applications need data processing with temporal constraints in their tasks; then one of the new challenges faced by wireless sensor networks (WSNs) is handling real-time storage and querying the data they process. Two main approaches to storage and querying data are generally considered warehousing and distributed. The warehousing approach stores data in a central database and then queries may be performed to it. In a distributed approach, sensor devices are considered as local databases and data are managed locally. The data collected by sensors must represent the current state of the environment; for this reason they are subject to logic and time constraints. Then, this paper identifies the main specifications of real-time data management and presents the available real-time data management solutions for WSNs, in order to discuss them and identify some open issues and provide guidelines for further contributions.

Contents

1. Introduction

Wireless Sensor Networks (WSNs) may be defined as a set of smart devices, called sensors, which are able to sense and transmit information about the environment on which they are deployed. These devices collect information for users interested in monitoring and controlling a given phenomenon and transfer them to a collected point called sink node. The latter make the information available to a gateway where the users can access via Internet. So as to obtain information, users use applications that communicate with the network through queries (Callaway, 2004; Sacks et al., 2003; Baronti et al., 2007). An illustration of a WSN may be found in Fig. 1.

This sort of network generally has a large number of nodes communicating and distributed on a given area to measure a physical quantity or an event monitoring. Each network node is considered intelligent and is equipped with an acquisition module, which provides a measure of environmental data (such as temperature, humidity, pressure, acceleration, sound, etc.), processing capacity, storage, communication, and energy. However, these resources are generally very limited, especially those of storage and energy, and the sensor nodes energy consumption is sometimes not negligible (Akyildiz et al., 2002, 2007).

Sensors can be used and placed everywhere they make information omnipresent. Consequently, systems based on sensor networks are more and more used in many areas providing then various types of WSNs (Mendes and Rodrigues, 2010). These numerous WSNs have allowed the development of many applications, which are generally connected to databases treating the amount of data collected from sensors. However, the processing

* Corresponding author. Tel.: +351275319891; fax: +351275319899.
E-mail addresses: ousmane.diallo@it.ubi.pt (O. Diallo), joeljr@ieee.org (J.J.P.C. Rodrigues), mbaye.sene@ucad.sn (M. Sene).

1084-8045/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2011.12.006

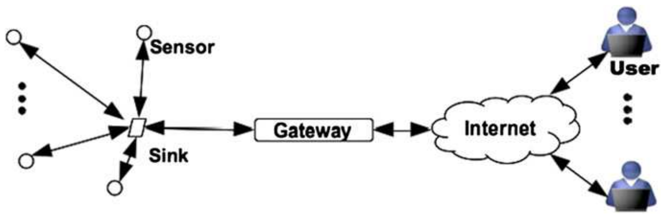

Fig. 1. Illustration of a wireless sensor network architecture.

time becomes increasingly critical for certain applications. These applications must query and analyze the data more quickly in order to make decisions and to react as soon as possible. Some examples of the most popular applications are the following: the control of network traffic (Cranor et al., 2002), transactional analysis (web, banking or telecommunication transactions) (Cortes et al., 2000), human motion tracking application (Chen and Ferreira, 2009), the tracking of actions on dynamic Web pages (Zhu and Shasha, 2002; Chen et al., 2000), monitoring of urban or environmental phenomena (Mainwaring et al., 2002; Ulmer et al., 2003), and the sensors data management (Arasu et al., 2003).

Once the sensors perform their measurement, the problem of data storing and querying arises. Indeed, the sensors have restricted storage capacity (Silva et al., 2004) and the ongoing interaction between network devices and environment results huge amounts of data.

There are two main approaches to data storage and querying in WSN: distributed and warehousing (Elnahrawy and Nath, 2004). In the first approach, researches recommend a distributed evaluation of requests on sensor networks, and aim at exploiting the capacities of calculation of sensors (Neto et al., 2008). The objective is to locally calculate in order to limit sending of messages, reducing thus the energy consumption. In the second approach, warehousing, one have a centralized system. Collected data from sensors (in stream) are sent to a central database server, in which user requests are processed. Note that this technique generates large data flows.

The data collected by the WSN must closely reflect the current state of the targeted environment. However, the environment changes constantly and the data are collected in discreet times. So, the collected data have temporal validity, as time advances they become less and less accurate, until the time where they do not reflect the state of the environment (Idoudi, 2009a, 2009b). It is fundamental that responses to application queries ensure that returned data comply with logic and temporal constraints. In this context, real-time data management on WSNs is necessary to dealing with those constraints. The main goal of real-time data management is to ensure temporal consistence data and process transactions within real-time constraints. The most relevant proposals of real-time data management on WSNs are going to be exposed along this work. This survey aims to study how real-time databases have been integrated in wireless sensor networks in order to satisfy real-time constraints of specifics critics applications connected to those sorts of networks. Moreover, a discussion and open issues on real-time data management on wireless sensor networks will be identified in order to make further contributions.

The remainder of this paper is organized as follows. Section 2 presents the various basic concepts and architectures used on real-time data storage and querying in WSNs while Section 3 exposes research contributions on real-time data management in WSNs. Section 4 discusses the techniques used on the studied approaches and proposes some research issues. Finally, Section 5 concludes the paper.

2. Background

The first purpose of a real-time system is the respect of the temporal constraints. For a database, this does not mean that the transactions execution must be fast, but that transactions must run in well-defined time intervals (fixed time) (Buffenoir, 2006; Lam and Kuo, 2001).

Like a traditional database management system (DBMS), a real-time DBMS (RT-DBMS) must process transactions and ensure that the logical consistency of the data is not violated. However, unlike a traditional DBMS, a RT-DBMS emphasizes on the temporal validity of the data and the time constraints or deadlines for transactions (Idoudi, 2009a, 2009b; DiPippo and Wolfe, 1997).

The main purpose of a RT-DBMS is to process transactions on time, while maintaining logical and temporal consistency of data. The temporal consistency expresses the need to maintain consistency between the current state of the targeted environment and the state as reflect by the database contents. The temporal consistency can be measured in two ways (Ramamritham, 1993):

- *Absolute consistency*, which deal with the need to maintain the view representing the state of the targeted environment consistent with the real state of the environment
- *Relative consistency*, which concerns data derived from other ones.

To satisfy these temporal constraints, the structure of the data must include these attributes: (*i*) timestamp, which indicates the instant when the observation relating the data was made; and (*ii*) absolute validity interval (*avi*) that denotes the time interval following the timestamp during which the data are considered valid. Another attribute can be considered; the imprecision, which refers to how the current state of the targeted environment may differ from the measured data (Ramamritham, 1993). The transactions also must have these attributes: (*i*) *liberation time* that represent the moment on which all the resources for the transaction processing is available; (*ii*) *computing time* that indicates the execution time needed for the transaction; and (*iii*) maximum time, which indicates the maximum time limit for the transaction execution and the periodicity that refers to the frequency with which the transaction happens (Chagas et al., 2010).

To satisfy the logical consistency of the data, transactions must be process with ACID (Atomicity, Consistency, Isolation and Durability) properties. But unlike the conventional databases, in real-time databases these properties are relaxed. Firstly, the atomicity may be relaxed. It is only applied to the sub-transaction that wants to deal with completely data consistency. Secondly, since timeliness is more important than correctness, in many situations, correctness can be traded for timeliness. Thirdly, the isolation allows transactions to communicate with others to better perform control functions. Similarly, in real-time databases, not all data must be permanent and some of them are temporal (DiPippo and Wolfe, 1997; Chagas et al., 2010; Ramamritham, 1993).

According to Ramamritham (1993), a set of data used to derive a new data item constitutes a *relative consistency* set. Each set R is associated with a relative validity interval denoted by R_{rvi}. Assume that, given a data item $d \in R$; d has a correct state if:

1. d_{value} is logical consistency, i.e., satisfies all integrity constraint.
2. d is temporally consistent:

- Absolute consistency: $(\text{current\_time} - d_{\text{timestamp}}) \le d_{\text{avi}}$
- Relative consistency: $\forall d' \in R,\ |d_{\text{timestamp}} - d'_{\text{timestamp}}| \le R_{\text{rvi}}$

According to Stankovic et al. (1999), a Real-time system can be seen as a *controlling system* and a *controlled system*. In an

Please cite this article as: Diallo O, et al. Real-time data management on wireless sensor network: A survey. Journal of Network and Computer Applications (2011), doi:10.1016/j.jnca.2011.12.006

automated manufacturing system, for example, the controlling system is composed by a computer and human interfaces that manage and coordinate the activities, while the controlled system is the manufacturing system with its robots, assembling stations, parts, and conveyers. Then, the controlled system represents the targeted environment. The controlling system interacts with its environment according to the data available, for example, provided by various sensors about the environment. The data collected by the WSN and perceived by the controlling system must closely reflect the current state of the targeted environment (Idoudi, 2009a, 2009b). However, the environment changes constantly and there are usually delay throughout the process of collecting, storing, and exploiting of the information characterizing the environment. This delay can generate huge inconsistent values that occurs bad prevision. This way, the real-time data management for WNSs can be a solution, since it provides several tools to deal with the temporal restrictions of data and transactions.

As the evolution of the traditional databases management systems, the sensor databases try to create an abstraction between the end-users and the sensor nodes thus allowing the users to only concentrate on the data that they need to be collected rather than bothering with the complexities of mechanisms deciding how to extract data from a network (Bonnet et al., 2001; Madden et al., 2002). This evolution of sensor databases has seen born various data storage and query management techniques designed specifically for WSNs. There are two main approaches to data storage and query in WSNs (Bonnet and Sheshadri, 2000; Bonnet et al., 2000): warehousing and distributed.

The warehousing approach (as illustrated in Fig. 2) is a centralized system (Bonnet et al., 2001). The data gathered by the sensors are sent to a central database where user queries are processed. In this case, the sensors act as simply collectors. The warehousing approach is the most used one in query processing. However, it has some drawbacks, such as, the huge number of generated data can easily create a bottleneck on the central server; the huge amount of information transferring waste the resources. The instructions processing is much less expensive than the wireless data transmission. Indeed, it is shown that power consumption to transmit a bit is equivalent to that consumed to carry out approximately 800 instructions (Madden et al., 2002). Moreover, it is clear that this approach is not adequate of no-historical queries because it involves time delay for the results.

In the other hand, the distributed approach (as illustrated in Fig. 3) exploits the capabilities of sensors calculation and the sensors act as local databases (Bonnet and Sheshadri, 2000). It aims to locally calculate the limit of sending messages in order to

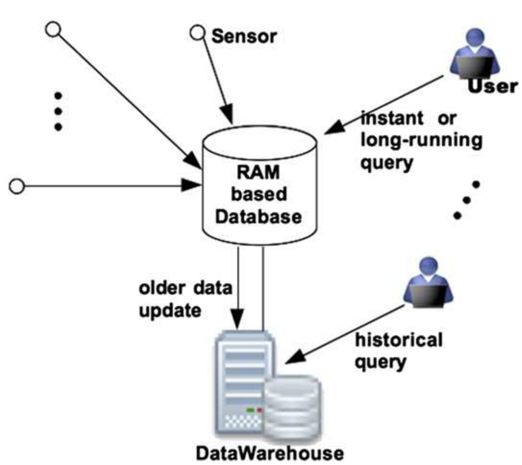

Fig. 2. Illustration of a warehousing approach.

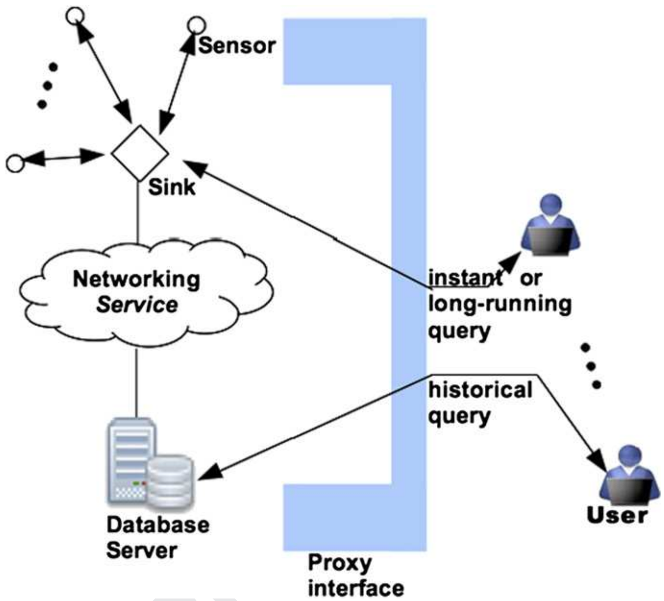

Fig. 3. Illustration of a distributed approach where sensor devices are part of the database.

save the energy consumption. In this approach, the data are stored in a central database server and in the devices themselves, enabling one to query both. Here, the devices act as part of the database. This approach can offer several advantages; on sensors, the processing of queries is done on *quasi* real-time; it supports long-running queries and instant queries; and the distributed approach increase the lifetime of the network. Although there is a significant advance in ubiquitous computing, the conditions for using sensor networks still restrict the use of data storage systems located within sensors. Indeed, the improvement of the sensors (operating time, improvement of calculation, and memory capabilities) does not cancel the risks of failure related to natural conditions.

There are three sorts of queries in sensor networks: (*i*) historical data queries, which are run against the server; (*ii*) instant queries, which are run against a device in an instant of time; and (*iii*) long queries, which refer to queries run against a device during a time interval (Bonnet and Sheshadri, 2000; Neto et al. (2004)).

3. Current solutions of real-time database techniques for WSNs

The real-time database data must represent the current state of the environment on which they were captured. This is particularly important especially for applications that monitor areas of risk. In this context, a good data structure for real-time access is very important. Thus, under volcanic monitoring, Noël et al. (2004a, 2004b), Noel et al. (2005a) have proposed a new sensor database spatiotemporal access method, named the Po-tree. Indeed, Po-Tree is an indexing system for centralized spatiotemporal databases, working on data from a fixed sensor network with real-time constraints. It distinguishes two characteristics of the data: spatial and temporal. The Po-tree structure is design as suit: the spatial characteristic indexed by a Kd-tree (Bentley, 1975) represent the first sub-tree and the temporal characteristic indexed through modified B+ – tree represent the second sub-tree. Each sensor is linked through this two linked sub-trees. In this tree, entries take the form 〈*left-pointer; point-position; link-to-temporal-tree; right-pointer*〉, representing thus

Please cite this article as: Diallo O, et al. Real-time data management on wireless sensor network: A survey. Journal of Network and Computer Applications (2011), doi:10.1016/j.jnca.2011.12.006

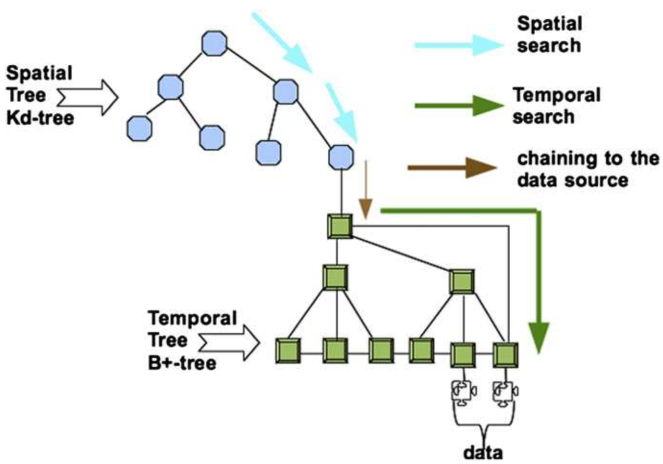

Fig. 4. Illustration of the spatiotemporal query with Po-Tree.

the spatial location, the position in the tree and a link to a temporal sub-tree. After the record of the measurement times a link is created to the recent measured values. In order to accelerate query processing of the most recent data; a direct link between the root and the last node, where the most recent data are usually recorded, is created. Queries are sent to the root of the spatial sub-tree that supports querying of different temporal sub-trees and provides the results. Indeed, a query is divided into two phases: if a query arrives, a first search is performed in the spatial sub-tree to determine the sub-trees of sensors concerned by the query. Subsequently, the spatial sub-tree forwards the query, having only temporal dimensions, to the sub-structures of the sensors involved and collects the results (ID sensor, measurements and time markers) to present to the user (see Fig. 4).

In real-time databases, in addition to logic constraints, data must also be consistent with the time constraints. So, it is necessary to efficiently store new arriving information into databases, to fast search of data and to manage the memory to avoid overload that can lead several miss deadline of transactions (Kang et al., 2004). This is particularly important for applications that need to query in real-time and access to the most recent data. To deal with these challenges, Firstly, Noel et al. (2005b), Noel and Servigne (2005) have proposed a real-time spatiotemporal indexing scheme for agile sensors, named the PasTree. The PasTree is an improved version of the PoTree. The structure is based to a multiversion spatial sub-tree; which keep tracks of the sensor location changes. Thus, each node keeps information about the past positions sensors and the actual ones. Moreover, PasTree includes two other access structures; the sensor tree for referencing sensor ID and the temporal object tree for each recorded sensor, allowing thus queries based on sensors identifiers or spatiotemporal characteristics. This proposal includes an interface located above the sub-trees of access and enables centralized management of queries. This interface allows hiding the user the multi-criterion access. If a query arrives, it redirects it depending on the type of access (spatial or identifiers criteria).

Secondly, Noël and Servigne (2006) have also proposed a third indexing method, named StH, for real-time management of main memory. The StH, spatiotemporal/Heat, is an indexing system for centralized spatiotemporal databases in RAM memory, working on data from a agile sensor network and taking into account the memory saturation of the database. The StH further allows multi-criteria access to the data. Indeed, it is based, on parity, in the construction of the PasTree: spatial sub-tree and identifiers sub-tree. In addition, it incorporates new sub-structures in order to prevent from the memory saturation of the database. Thus, in addition to spatial and identifiers sub-trees, each sensor is associated with a dedicated sensor sub-structure called staircase because of its shape. This dedicated sub-structure contains all information relative to a sensor (id, history of positions and links to the measures). This index allows the resolution of queries based on spatial criteria or sensor identifiers. The management of the memory saturation is done by using a heat-function associated with each staircase, i.e., each sensor. Thus, the coldest data are regularly transferred from the sensor stairs to the data warehouse. This transfer is triggered after the account of a maximum number of data input by the sensor. Less hot data correspond to the less important data the selection of these less important data is made by some criteria, for example by priority.

Servigne and Noel (2008) have shown the complete structure PoTree, PasTree and StH but this time it is increasing the StH linking the database (into memory hand) to a data warehouse. They emphase the apects of transfer of less important data toward a data warehouse, to release the database. Thus, they based on a warehousing approach where after collecting, data are sent toward a central point (in main-memory databases) for real-time querying and analysis. For long-running querying and free database memory, data are sent toward a disk or secondary-memory (warehousing).

The data acquired by the sensors must represent the current status of the targeted phenomenon. However, this status is subject to constant changes. Thus, in (Bouju et al., 2009). Noel et al. (2005b), the authors attempt to take into consideration the location of the sensors; which can be fixed, agile and mobile. Based on the UML formalism, they define a spatiotemporal sensor data model for real-time storing according to the position (fixed, agile, mobile) of the sensors. Moreover, they emphasis of the importance of metadata modeling and present a formalism of object issued by the sensors.

The data stream management systems meet the operating requirements of applications to query data stream generated continuously flows (Golab and Özsu, 2003). Generally, these types of applications are subject to a huge amount of stream data, triggers, imprecise data, and real-time requirements. To deal with this challenge, in (Abadi et al., 2003; Carney et al., 2002), the authors have proposed a system that provide real-time data stream processing capabilities with a real-time metric the average latency of data tuples, named Aurora. Aurora is a data-flow system that process incoming stream according to the requirements of the applications (see the aurora architecture in Fig. 5). It includes a set of operators for satisfying the stream processing requirements. Each operator consumes data input, performs operations, and produces results in a continuous manner. Among these operators we can note: windowed operators, filter operator, etc. Aurora can process continual queries in real-time processing according to QoS specifications, but unlike the relational approach, in Aurora query building is procedural, i.e., a query is created using a graphical interface where one place boxes that

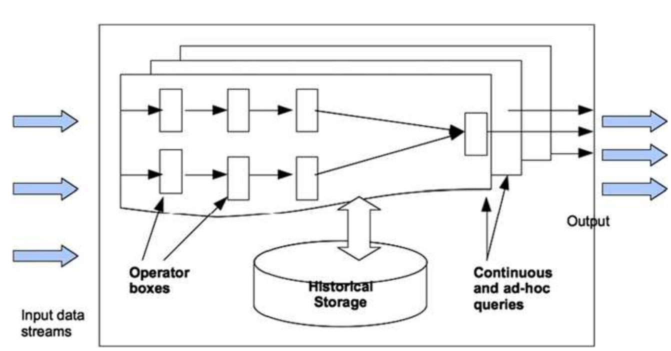

Fig. 5. Architecture of Aurora.

Please cite this article as: Diallo O, et al. Real-time data management on wireless sensor network: A survey. Journal of Network and Computer Applications (2011), doi:10.1016/j.jnca.2011.12.006

represent transactions and arrows that represent the data stream. Moreover, the evaluation is made tuple by tuple, unlike the approach of traditional databases where queries are evaluated on relations. To deal with the QoS, Aurora uses a load shedder mechanism that, if it detects an overload situation, flash load till the system find a good performance. Moreover, it uses several utility functions in QoS and associates QoS curves with outputs from stream processing to support continuous timing requirements.

Considering the real-time approach, the main idea is to answer queries in their time constraints; real-time applications need predictable responses. Queries should be process in accord with their deadline. So, RTSTREAM (Wei et al., 2006) try to comply with this constraint in huge amount of sensor data stream. RTSTREAM is a real-time data stream management system (DSMS) that supports a real-time data stream query model, named PQuery. The PQuery model improves upon the Continuous Query Language (CQL) (Arasu et al., 2006) as it allows applications to specify the query frequencies and deadlines; which in turn is used to deal with the queries time constraint. Instead of always-reinitiated query instance by the incoming data stream (like in others continuous query model), PQuery model does not allow the change of the input during the course of the instance execution, i.e. if a query is registered, its instances are periodically initiated by the DSMS system and a newly arrived data stream are processed by the next query instance. Moreover, RTSTREAM includes an overload protection mechanism named data admission for better deal with query deadline miss ratio. The data admission uses sampling and shedding strategies to reduce the incoming data volume during periods of high load. This system produces periodic responses in real time without interrupting the query instance execution, but it does not provide the latest result of the incoming data stream because of the incoming data stream reported to next query instance.

The real time context involves both taking into account the time constraints that a good quality of information provided (Ramamritham, 1993). Thus, the definition of metadata in real time context for spatiotemporal sensor data can be a good support, particularly in decision making-process. There is a standard that represents the structure of the spatial metadata: The ISO 19115 standard (ISO19115, 2006; Servigne et al., 2006) but this standard does not include the dynamic objects and real-time aspect. To meet this need, researchers in (Gutierrez and Servigne, 2007; Gutierrez et al., 2007) attempt to gather the basic features of spatiotemporal data and metadata to better assist in the management quality of real-time spatiotemporal data. Thus, referred to the nature of the sensors; their characteristics and their positions, they define a sensor (fixed, mobile or agile) spatiotemporal data model in real time context. According to the sensor data model and the future function of the metadata, they expose a generic metadata that can be use by any real-time spatiotemporal application and in the other hand a specific metadata for specific application. They also define two metrics that can be help to differentiate data to metadata: the data semantic; characterized by their final use and the data dynamicity characterize by the variation of values on time.

The definition of spatiotemporal metadata in real time is well appreciated for good quality of service in real-time context, but must be accompanied by good update management, extraction and exploitation policy, especially in dynamic and real-time context. Sensor networks generate large amounts of data very variable and the access time is critical, especially in the context of real time.

Unlike the researches that focus in warehousing approach, in real-time databases for sensor networks (Neto et al., 2004), the authors use a distributed approach to propose an integration between real-time database technology and sensor network systems. Thus, they try to perform both logical and timing verification for the data and transactions. In their model, sensors acquire data; store and periodically they make updates to the server. Thus, the long-run queries and instant queries can now run against the sensors that are programmed to perform the temporal constraints of data and transactions while the historical queries will be run against the server. For handling the execution concurrence of the transactions in conflict, they propose an in-network layer that deals with the serializability property. Moreover, unlike the other proposals that use programming models to assess their performance, their proposed architecture (sensor network, database server and their components) is modeled by a Colored Petri nets formalism in order to analyze the system performances. They try to demonstrate that their model is consistent with the logical and temporal constraints of data and transactions in the context of real-time data management.

Like a conventional database management systems, a real-time database management system must process transactions while ensuring that the database consistency is not violated. Thus, Chagas et al. (2010) present some real-time database management techniques to deal with this challenge. It uses a distributed approach where the network devices act as database and periodically send the acquired data to the database server. To deal with time constraint, the devices and the server include a program that process transaction while handling the time constraints of the data and transactions. The PostgreSQL DBMS is used to store data provided by the sensors and the Query Language for Real-Time Databases (QL-RTDB) (Leite, 2005) is used to access to both the server and the devices via an application interface. To deal with the eventual concurrent transactions an algorithm that takes into en consideration the time constraints.

Due to the fact that the real time applications are subject to a very huge amount of data to process and are temporal constraints many transaction may miss their deadline, degrading then the performances. In these applications, it is also important to access fresh data that effectively reflect the current status of the targeted environment. Considering these problems, Kang et al. (2004) have proposed a real-time main memory database architecture, named QMF (QoS management architecture for deadline Miss ratio and data Freshness) to improve the quality of service (QoS). This proposal attempt to balance the deadline miss ratio compared with data freshness in considering the applications requirements. Indeed, this model allows specifying the desired miss ratio and data freshness for a specific application. To deal with the miss ratio, QMF use a feedback controller. It includes a controller that periodically measures the miss ratio, calculates the error miss ratio, i.e, the difference between the values of miss ratio desired and the actual measured value and react to correct the error. The QMF also includes a freshness manager that updates more or less sensor data on demand according to the miss ratio control messages and the workload. Moreover, QMF make control admission to incoming transactions to decrease the overload situations. In order to balance potentially conflicting miss ratio and freshness requirements, it uses a flexible method. Thus, it uses a range of quality of data (QoD) that is compared to the sensor data in order to accept their freshness, if necessary. This can be relaxed the update periods of sensor data and decrease the update workload in case of overload. Moreover, the freshness of sensor data is maintained according to flexible validity intervals.

Cross-layer solution states that parameters of two or more layers can be retrieved and/or changed in order to achieve an optimization objective (Mendes and Rodrigues, 2010). The cross-layering concept has been first proposed for TCP/IP networks, when wireless links were deployed (Srivastava and Motani, 2005). Many optimization solutions have been based on layers. In this

Please cite this article as: Diallo O, et al. Real-time data management on wireless sensor network: A survey. Journal of Network and Computer Applications (2011), doi:10.1016/j.jnca.2011.12.006

context and in the case of a human motion tracking application (Chen and Ferreira, 2009), Gonçalves et al. (2009) have proposed an architecture based on layers in order to optimize the delay of the sensor data dissemination. Thus, this architecture deals with real-time data dissemination over Internet. It is a four tiers architecture that divide the data dissemination process into four different layers: the WSN Layer for the data acquisition, the Data Layer for the presentation, the Transport Layer for the transmission and the Client Layer for the consumption (see Fig. 6). The WSN Layer collects the data, some times with low level processing, and sends it to the Data Layer. The Data Layer acts as a gateway that interconnects the WSN with others networks. When the Data Layer receives the data from the WSN Layer, it encodes it, in XML form for example, and sends it to the Transport Layer. The Transport Layer, on its round, with time constraints, establishes a single or multiple data transport channel and forwards the data to the Client layer. Finally, the Client Layer receives the data packets and makes some processes in order to improve the service. The Transport Layer use two specifics transport protocols: the Extensible Messaging and Presence Protocol (XMPP) and the Real-Time Protocol (RTP) (Schulzrinne et al., 2003). To deal with real-time transmission constraints, the RTP is used for communication between the Data Layer and the Transport Layer and between the Transport Layer and the Client Layer, in that it provides an improved of the delivery arrival time with UDP protocol. The Jingle XMPP extension is used to establish sessions between the Data Layer and the Transport Layer or between Client Layer and the Transport Layer.

Replication (Plattner and Alonso, 2004) allows managing multiple copies that differ at a given time, but eventually converging to the same values (Gardarin and Gardarin, 1997). The motivations to make a replication are essentially improved performance and increased data availability. Thus, in order to improve the access time to sensor data at any node and to ensure scalable propagation of updates over the network, Mathiason et al. (2008) have proposed a communication scheme in sensor networks using a distributed real-time database with Virtual Full Replication. This scheme is called virtual full replication with adaptive segmentation for sensor networks (ViFuR-ASN). It based on a two-tier approach composed by sensor tier nodes and database tier nodes, where the database tier has more powerful nodes and greater energy supplies. As depicted in Fig. 7, the sensor nodes connect to a suitable database node, forming thus a sub-network, reach the database node by multi-hops and update theirs corresponding readings (sensor data objects) in the database. Each database node has a virtual fully replicated database (an image of full replicated to the database), which stores the data objects belong to the sensor nodes connected to it. The clients in moving can select the database node they want to connect in order to access to sensor data objects. Thus, the search space of client nodes for sensor data is thus limited to the number of database nodes. In the database tier, the database nodes replicate the updates in the same manner. The database nodes are composed, among others important modules, the Segmentation

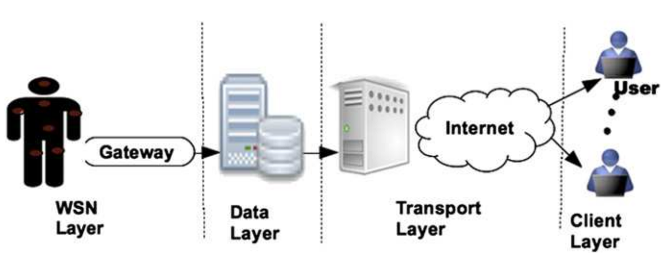

Fig. 6. Illustration of the system architecture.

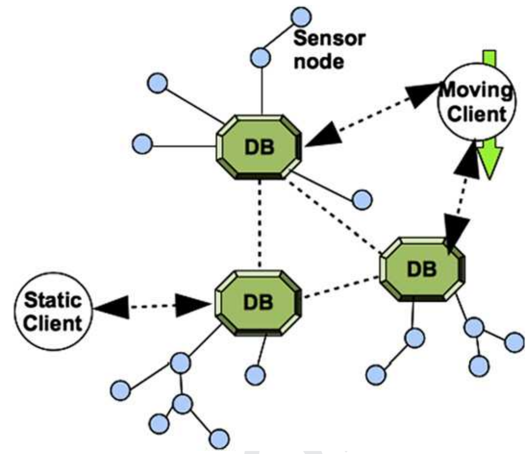

Fig. 7. Illustration of the architecture, using a distributed real-time database with Virtual Full Replication (ViFuR-ASN).

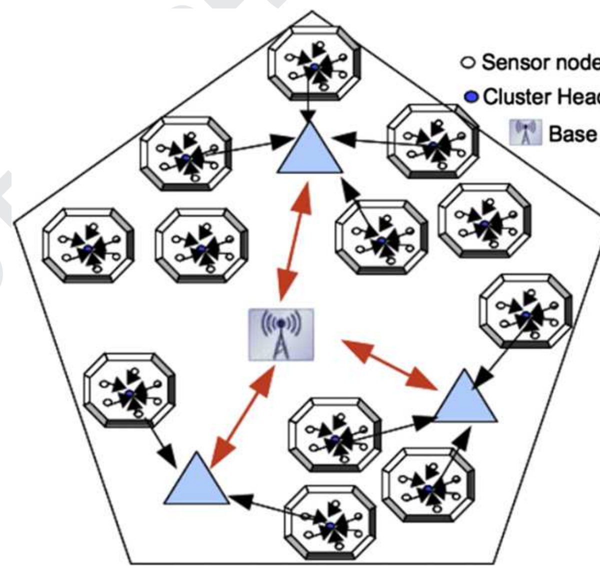

Fig. 8. Illustration of the different components.

Manager that manages the allocations of replicas in the database; the Updater Logger that manage to send local updates to others nodes with the same replicas of the updates data objects; and the Transaction Manager, which executes the transactions. Each sensor objects has unique identifier and the location of each sensor is stored as a separate database, allowing thus spatial queries. To deal with timely execution of transactions and the network latency, ViFuR-ASN adopts the same independent updates, virtual full replication, conflict detection and resolution approaches of DeeDS (Andler et al., 2007), but unlike it, ViFuR-ASN uses locally replication that makes values always available and reduces thus the eventual delay. This is also allows predictable local access time.

Gupta and Dave (2008) have proposed an architecture based on clusters in order to handle a reliable and real-time placement and dissemination of data in WSNs. As depicted in the Fig. 8, the clusters and cluster-heads (CH) are formed in arbitrary manner and these roles are alternated to better distribute energy use in the network. In each cluster, the sensors communicate data to their CH that aggregates data and reduces thus the size of data to be transmitted to the base station via Action and Relay Stations

Please cite this article as: Diallo O, et al. Real-time data management on wireless sensor network: A survey. Journal of Network and Computer Applications (2011), doi:10.1016/j.jnca.2011.12.006

(ARS) to prevent excessive dissipation of energy. Also, to timely perform actions, the ARS sends data to sink nodes via other ARS. For improving the storage and access of the data, the data centric storage mechanism is used. Also, the data stored in a base station is replicated in the adjacent base stations in order to increase the availability of data and to speed the query processing. For interaction in the network, the queries used are SQL-like queries.

The following tables (Tables 1 and 2) provide a summary to get an overview of the proposed solutions for real-time database management on WSNs.

4. Open issues

The real-time data management for WSNs can be seen as all the necessary resources to meet requirements (storage and exploitation) of real-time applications working on data from sensors. These real-time applications differ from other applications by taking into account time constraints, compliance with which is as important as the accuracy of the result. In other words, these applications should not just deliver accurate results, but within the deadlines. Building mechanisms to meet these

Table 1
Summary to various kinds of proposals and their features (part 1).

Name of Project/ Mechanism/ Authors	**Problematics**					
	Type of approach	Basic concepts	Queries expression	Queries evaluation/ metrics	Types of application	Platform
PoTree, 2004	Centralized approach	Real-time spatiotemporal data indexing for fixed sensors	Spatiotemporal queries: spatial criteria for designating a sensor or zone; interval or time instant for the temporal aspect	Real-time update; query on most recent data in RAM Memory	Monitoring applications	Simulation
PasTree, 2005	Centralized approach	Real-time spatiotemporal data indexing for agile sensors	Spatiotemporal queries: ID or spatial criteria for designating a sensor or zone; interval or time instant for the temporal aspect (multi-criteria queries)	Real-time update; query on most récent data in RAM Memory	Monitoring applications	Simulation
StH (SpatioTemporal-Heat índex), 2006	Centralized approach: derived from PasTree	Real-time spatiotemporal data indexing for agile sensors; management of memory saturation	Spatiotemporal queries: ID or spatial criteria for designating a sensor or zone; interval or time instant for the temporal aspect (multi-criteria queries)	Real-time update; query on most récent data in RAM Memory; Memory saturation	Monitoring applications	Simulation
Bouju et al., 2009	Centralized approach	Spatiotemporal sensor data model for real-time storing	–	–	–	–
Aurora, 2002, 2003	Centralized approach	Data Stream management system	Procedural queries	Average latency	Monitoring applications	–
RTSTREAM, 2006	Centralized approach	Real-time data stream management to deal with deadline of periodic queries over data stream	PQuery: SQL-like language	Deadlines of queries	Real-time application based on data streams	Simulation
Gutierrez and Servigne, 2007; Gutierrez et al., 2007	Centralized approach	Real-time metadata for real-time management of spatiotemporal data	–	Real-time spatiotemporal metadata	Monitoring applications	–
Neto et al., 2004	Distributed approach	In-network processing, concurrency control	Message sending	Logical and timing constraints for data and transactions	–	Analytic simulation
Chagas et al., 2010	Distributed approach	In-network processing, algorithms to negotiate between logical and temporal consistency	SQL-like language	In-networking processing; Logical and timing constraints for data and transactions	Real-time applications	Simulation

Table 2
Summary to various kinds of proposals and their features (part 2).

Name of Project/ Mechanism/ Authors	**Problematics**					
	Type of approach	Basic concepts	Queries expression	Queries evaluation/metrics	Types of application	Platform
QMF, 2004	Centralized approach	Transactions exécution within their deadlines using fresh data	–	Deadline miss ratio and data freshness metrics	Real-time application	Simulation
Gonçalves et al., 2009	Distributed approach	Real-time sensor data dissemination over Internet based on a four tiers architecture	Text Messages sending	Services applying ; minimizing the transmission delay	Human motion tracking application	–
Mathiason et al., 2008	Distributed approach	communication scheme in sensor networks using a distributed real-time database with Virtual Full Replication	–	In-netwroking exécution ; minimizing the delay of latency ; disponibility; Scalability	–	Simulation
Gupta and Dave, 2008	Distributed approach	Real-time in-networking data storage (replication) and dissemination	SQL-like queries	In-networking processing; Real-time data placement, availability, energy saving	Real-time application	–

Please cite this article as: Diallo O, et al. Real-time data management on wireless sensor network: A survey. Journal of Network and Computer Applications (2011), doi:10.1016/j.jnca.2011.12.006

kinds of applications is no small matter. Indeed, the techniques previously used in traditional databases are difficult to apply in sensor systems because of their limited resources. Although efforts have been made to deal with these constraints, there are still many challenges that must be addressed to reach best solution. Some identified challenges are given below:

First, in these real-time systems, the time constraint is very delicate especially for long-running queries and instant-queries. Most of the proposals are based on the warehousing approach, which, among other drawbacks, can cause a delay on the response times. On the other hand, others proposals focus on the distributed approach that seems, based on its access mode (direct access to sensor nodes), provide little help on the time constraint, however with some risk because there may be a sudden failure of sensors (according to the unstable environment). This may lead to a delay or lack of information that influences the analysis time or even on the blocking of the system. Hence, one can opt for a hybrid approach which allowed one or the other solution depending on the requirements of the application that is the source of the query. This solution will take into account both the time constraints that the risk of failures. Moreover, the connection of the wireless sensor network and Internet will probably cause to the sensor system to be subject to a huge amount of queries. So, system must take account to the scalability.

In data management systems, the inclusion of meta-data that give the characteristics of the data is essential. Although some proposals include it in their solutions it is not enough. Thus, these solutions may be revised and improved for best metadata management in order to have fast and accurate routing.

Another thing to consider is the management of simultaneous queries. Indeed, systems connected to a database are most often subject to numerous queries especially for consultation. Hence, a good policy of multiple queries optimizing would be a good deal.

Finally, in real time systems, for some applications, the accuracy of results may be sacrificed to reduce the response time. Thus, one can think of an approach that allows for an approximation of the queries response.

5. Conclusions and future work

Sensor systems offer new opportunities for building applications that have information and control in areas previously very difficult or even inaccessible. After the acquisition of the information, a major problem of data storage and exploitation arises, particularly for system that dealing with real-time. The researchers have more contributed in the construction of the specific building which is not easy because the data management techniques used in traditional databases are not generally suitable for sensor networks because of their specificities.

This work has reviewed the various solutions for managing sensor data in real-time. To this end, a survey of the oldest to the most recent proposals in this area has been done and it is particularly interested in various stages of data storage, processing and optimization of real-time queries required by the real-time applications.

Many proposals are based on the warehousing approach, considering all treatments in centralized database. Other proposals use a distributed approach, considering the sensor nodes as part of the database. Both have advantages and disadvantages, as depicted earlier.

At the end of this analysis, one can say that a solution that optimize the real-time query processing while ensuring data availability in case of failure is very important in the management of sensor data in real time. Thus, a hybrid approach can help to deal with these two challenges. Furthermore, for the time being, there is no proposal that has built-in facilities for approximate query answering. The design of a framework that takes into account these specificities can well meet the requirements to real-time applications based on WSNs.

In future works, we will use a high level modeling such as Stochastic Well-formed Petri Nets (SWN) to study and analysis this Wireless Sensor Network. SWN is a powerful and accurate tool for modeling complex systems with concurrency, synchronization and cooperation. Refined performance indices of WSN (real-time query processing, availability and response time of sensors) can be obtained by simulation or by exact computing.

Uncited references

Altisen et al. (2005), Costa et al. (2010), Fernendes (2005).

Acknowledgments

Part of this work has been supported by the *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Portugal, and by National Funding from the FCT – *Fundação para a Ciência e a Tecnologia* through the PEst-OE/EEI/LA0008/2011 Project.

References

Abadi DJ, Carney D, Çetintemel Ugur, Cherniack M, Convey C, Lee S, et al. Aurora: a new model and architecture for data stream management. The VLDB Journal (Springer) 2003;12(2):120–39.

Akyildiz FI, Melodia T, Chowdhury RK. A survey on wireless multimedia sensor networks. Journal of Computer networks (Elsevier) 2007;51(4):921–60.

Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor networks: a survey. Computer networks (Elsevier) 2002;38:393–422.

Altisen K, Bouyer P, Cachat T, Cassez F, Gardey G. Introduction au contrôle des systèmes temps-rèel. Journal européen des systèmes automatisés 2005;39:367–80.

Andler FS, Brohede M, Gustavsson S. DeeDS NG: Architecture, design, and sample application scenario. *Handbook of Real-Time and Embedded Systems', CRC Press. To appear in June*; June 2007.

Arasu A, Babu S, Widom J. The cql continuous query language: Semantic foundations and query execution. The VLDB Journal—The International Journal on Very Large Data Bases (Springer-Verlag New York, Inc.) 2006;15(12):121–42.

Arasu A, Babcock B, Babu S, Datar M, Ito K, Motwani R, et al. STREAM: The Stanford Stream Data Manager. IEEE Data Engineering Bulletin. Citeseer 2003:19–26.

Baronti P, Pillai P, Chook CVW, Chessa S, Gotta A, Hu FY. Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards. Journal of Computer communications (Elsevier) 2007;30(7):1655–95.

Bentley LJ. Multidimensional binary search trees in database application. Journal of IEEE Transaction on software engineering (IEEE) 1975:4.

Bonnet Ph., Seshadri P. Device Database Systems. In: *Proceedings of the International Conference on Data Engineering ICDE'99*. San Diego, CA: IEEE Computer Society; 2000.

Bonnet Ph, Gehrke J, Seshadri P. Querying the Physical World. Journal of Personal Communications, IEEE 2000;7(5):10–5.

Bonnet Philippe, Gehrke Johannes, Seshadri Praveen. Towards sensor database systems. *In MDM'01: Proceedings of the Second International Conference on Mobile Data Management*. London, UK: Springer-Verlag; 2001. p. 3–14.

Bouju A, Bertrand F, Servigne S, Devogele T, Ray C, Martin H. Gestion de données spatio-temporelles au sein de bases de données capteurs. *Conférence sur les Technologies de l'Information, de la Communication et de la Géolocalisation dans les Systémes de Transports.* 2009.

Buffenoir Christophe. Petite introduction théorique au temps réel. GNU Linux Magazine France 2006;82:76–9.

Callaway EH. Wireless sensor networks: architectures and protocols.CRC Press; 2004.

Carney D, Centintemel U, Cherniack M, Convey C, Lee S, Seidman G, et al. Monitoring streams – a new class of data management applications. In: Proceedings of the 28th international conference on Very Large Data Bases. VLDB Endowment, 2002. p. 215–26.

Chagas Lunas Dantas, Lima Eduardo Pereira, Ribeiro Neto Pedro Fernandes. Real-Time Databases Techniques in Wireless Sensor Networks. *Sixth International Conference on Networking and Services.* IEEE, 2010. p. 182–7.

Chen J, Ferreira L. Human Motion Tracking using Wireless Sensor Networks. Technical Report HURRAY-TR-090407, available online from 〈www.cister.isep.ipp.pt〉, 2009.

Chen J, DeWitt David J, Tian F, Wang Y. NiagaraCQ: A scalable continuous query system for internet databases. *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. New York: ACM Press; 2000. p. 379–90.

Cortes C, Fisher K, Pregibon D, Rogers A. Hancock: a language for extracting signatures from data streams. *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York: ACM Press; 2000. p. 9–17.

Costa CM, Leite CRM, Neto FMM, Neto PFR. AMSO: REAL-TIME SYSTEM FOR OIL WELLS ON-SHORES USING SENSORS NETWORK. *The 5th Euro American Conference on Telematics and Information Systems (EATIS 2010)*. Panama, 2010.

Cranor C, Gao Y, Johnson T, Shkapenyuk V, Spatscheck O. Gigascope: High performance network monitoring with an SQL interface. *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*. New York: ACM Press; 2002. p. 623.

DiPippo LC, Wolfe VF. Real-time databases. Journal of Database Systems Handbook, Multiscience Press (Citeseer) 1997.

Elnahrawy Eiman, Nath Badri. Context-Aware Sensors. Wireless Sensor Networks (Springer) 2004;2920:77–93.

Fernendes YYMP. Técnica de Controle de Concorrência Semântico para Sistema de Gerenciamento de Banco de Dados em Tempo-Real. Dissertação, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Campina Grande, Campina Grande, 2005.

Gardarin O, Gardarin G. Le Client-Serveur.Paris: Editions Eyrolles; 1997.

Golab Lukasz, Özsu Tamer M. Issues in data stream management. ACM Sigmod Record (ACM) 2003;32(2):5–14.

Gonçalves Joel, Ferreira Luis Lino, Chen Jianxin, Pacheco Filipe. Real-Time Data Dissemination for Wireless Sensor Networks using XMPP. 2009.

Gupta S, Dave M. Real time approach for data placement in wireless sensor networks. International Journal of Electronic Circuits System 2008;2(3):132–9.

Gutierrez C, Servigne S. Métadonnées spatio-temporelles temps réel. Journal de INGENIERIE DES SYSTEMES D INFORMATION (HERMES SCIENCE) 2007;12(2):97–119.

Gutierrez Rodriguez C, Servigne S, Laurini R. Towards real time metadata for network- based geographic databases. *Proceedings of ISSDQ2007, 5th International Symposium of Spatial Data Quality* 2007. p. 13–15.

Idoudi Nizar, Duvallet Claude, Sadeg Bruno, Bouaziz Rafik, Gargouri Faiez. How to model a real-time database?. *Proceedings of 12th IEEE International Symposium on Object-oriented Real-time distributed Computing (IEEE ISORC'2009)*. Tokyo, Japan: IEEE, 2009a. p. 321–5.

Idoudi Nizar, Louati Nada, Duvallet Claude, Bouaziz Rafik, Sadeg Bruno, Faiez Gargouri A. Framework to Model Real-Time Databases. International Journal of Computing & Information Sciences 2009b;7:1–11.

ISO19115. *An International Metadata Standard for Geographic Information*. 〈http://grdc.bafg.de/servlet/is/2376/〉. 2006; accessed March 2011.

Kang Kyoung-Don, Son Sang H, Stankovic John A. Managing deadline miss ratio and sensor data freshness in real-time databases. Journal of IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING. IEEE 2004:1200–16.

Lam YK, Kuo WT. Real-Time Database Systems: Architecture and Techniques.-Netherlands: Springer; 2001.

Leite CRM. Linguagem de Consulta para Aplicações em Tempo-real. Dissertaço, Programa de Pós-Graduação em Engenharia Elétrica.Campina Grande: Universidade Federal de Campina Grande; 2005.

Madden Samuel, Franklin Michael J, Hellerstein Joseph M, Wei Hong. TAG: A tiny aggregation service for ad-hoc sensor networks. *In OSDI*. Citeseer, 2002.

Mainwaring A, Culler D, Polastre J, Szewczyk R, Anderson J. Wireless sensor networks for habitat monitoring. *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. New York: ACM Press; 2002. p. 88–97.

Mathiason G, Andler FS, Son HS. Virtual full replication for scalable and adaptive real-time communication in wireless sensor networks. *The Second International Conference on Sensor Technologies and Applications*. IEEE, 2008. p. 55–64.

Mendes LDP, Rodrigues JJPC. A survey on cross-layer solutions for wireless sensor networks. Journal of Network and Computer Applications; Elsevier 2010.

Neto Ribeiro PF, Fernandes YYMP, Mendes Neto FM, Leite CRM, Costa CM. Rede de Sensores: O Estado da Arte. *EPOCA – Escola Potiguar de Computação e suas Aplicações*. 2008. p. 1–16.

Neto Pedro Fernandes R, Perkusich Maria LB, Perkusich Angelo. Real-Time Database for Sensor Networks. *In: Proceedings of the 6th International Conference on Entreprise Information Systems*. 2004. p. 599–603.

Noel G, Servigne S. Indexation multidimentionnelle de bases de donnees capteur temps-réel et spatiotemporelles. Journal of INGENIERIE DES SYSTEMS D INFORMATION (HERMES SCIENCE) 2005;10(4):59–88.

Noel G, Servigne S, Laurini R. Spatial and Temporal Information Structuring for Natural risk monitoring. *Proceedings of the GIS Planet*. Lisbonne, 2005b.

Noel G, Servigne S, Laurini R. The Po-tree: a Real-time Spatiotemporal Data Indexing Structure. Journal of Developments in Spatial Data Handling (Springer) 2005a. 259–27.

Noël Guillaume, Servigne Sylvie. Structuration de données spatio-temporelles temps-réelles: vers la gestion de la saturation de base de données. *In: Proceedings of INFORSID 2006*. Tunisie, 2006.

Noël Guillaume, Servigne Sylvie, Laurini Robert. Real-time spatiotemporal data indexing structure. *Proceeding of 2004 AGILE 7th Conference on Geographic Information Science*. Heraklion: Citeseer, 2004a. p. 261–8.

Noël Guillaume, Servigne Sylvie, Laurini Robert. The Po-tree: a soft real-time spatiotemporal data indexing structure. *Developments in Spatial Data Handling. SDH04, 11th International Symposium on Spatial Data Handling*. Springer Verlag. 2004b. p. 259–70.

Plattner C, Alonso G. Ganymed: Scalable replication for transactional web applications. Journal of Middleware 2004 (Springer) 2004:155–74.

Ramamritham K. Real-time databases. Journal of Distributed and Parallel Databases (Springer) 1993;1(2):199–226.

Sacks L, et al. The development of a robust, autonomous sensor network platform for environmental monitoring. Sensors and their Applications XXII (Citeseer) 2003.

Schulzrinne H, Casner S, Frederick R, Jacobson V. RFC3550: RTP: A Transport Protocol for Real-Time Applications 2003.

Servigne S, Noel G. Real time and spatiotemporal data indexing for sensor based databases. Journal of Geospatial Information Technology for Emergency Response (Taylor & Francis Group) 2008;6:123–42.

Servigne S, Lesage N, Libourel T. Spatial data quality components, standards and metadata. Spatial data quality: an introduction. International scientific and technical encyclopedia 2006:179–208.

Silva FA, Braga TRM, Ruiz LB, Nogueira JMS. Tecnologia de Nós Sensores Sem Fio. Contrôle & Instrumentação; agosto 2004;92:76–87.

Srivastava V, Motani M. Cross-layer design: a survey and the road ahead. Journal of Communications Magazine, IEEE (IEEE) 2005;43(12).

Ulmer C, Yalamanchili S, Alkalai L. Wireless distributed sensor networks for in-situ exploration of mars. *Work in Progress for NASA Technical Report* (Citeseer), 2003.

Wei Y, Son SH, Stankovic JA. RTStream: Real-Time Query Processing for Data Streams. *IEEE Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'06)*. Gyeongju, Korea: IEEE Computer Society; 2006.

Zhu Yunyue, Shasha Dennis. Statstream: Statistical monitoring of thousands of data streams in real time. *Proceedings of the 28th international conference on Very Large Data Bases*. VLDB Endowment, 2002. p. 358–69.