N-way Local SemiJoin : A Filtering Technique for N-Way Joins in Wireless Sensors Networks

Djail boubekeur, Hidouci Walid-Khaled, Loudini Malik
ESI
Algeria
{b_djail@esi.dz}, {w_hidouci@esi.dz},{m_loudini@esi.dz}

ABSTRACT: *In wireless sensor networks, the database consists of data collected by the different network nodes. This database can be accessed by relational queries, such as: selections, projections and joins. Joins are queries that use a lot of energy, which is a determining factor for network lifetime. In this paper, a new technique for n-way join queries is suggested, using an in-network processing and adopting the filtering approach. Simulation results showed very good performances of the proposed technique compared with a central processing of those queries at the sink.*

Keywords: Wireless Sensors Networks, Cost of Communication, In-network Join, N-way Join

Received: 22 September 2015, Revised 29 October 2015, Accepted 2 November 2015

1. Introduction

A wireless sensor network is composed of a very large number of inexpensive and low energy nodes, with processing and memorizing capabilities. The nodes are deployed in a region of interest; they use radio frequencies as communication channels to provide data collection and dissemination.

A sensor is considered as a data source which generates records at different reception actions.

The records of nodes of the same type in the same network compose together a distributed table. The sensors network could be considered as a distributed database system [Yao 03].

Collecting received data could be expressed as relational queries, such as : selections, projections, union operations, aggregation and joins.

In sensors network, data transmission consumes more energy than data processing in the node [Zhao 04]. It is important to reduce the quantity of data transmitted during a join operation in order to reduce the energy consumed by the sensors.

The techniques proposed recently such as : Distributed-Broadcast of Coman and al. [Coman 07], Mediated Join of Pandit and

Gupta [Pandit 06], Synopsis Join of Yu and al [Yu 06], and INJECT of Min and al [Min 11], have been limited to binary joins queries. N-way joins (joins between more than two tables) are rarely addressed because they can be implemented by a succession of binary joins. However, this approach can lead to excessive energy consumption, especially in cases where the intermediate results generate large tables.

A more detailed outline of some of these techniques is given in [Djail 13][Djail 15] presenting a state of the art of the techniques used in binary joins in the context of wireless sensors networks.

The n-way joins are of a remarkable interest in the networks of sensors. A practical example of the application of this type of joins is the traffic control of the vehicles on trajectories made up of several (more than two) zones covered by wireless sensors.

Monitoring bird's migrations, through several geographical areas is another example of application of this type of joins.

This paper presents a technique for an execution of n-way joins in wireless sensor networks.

An overview of the join queries in wireless sensor network is given first.

Then, a state of the art is presented and a description of the proposed technique is performed.

Finally, an analysis of the performance and future prospects are presented.

2. General Characteristics of Joins Operations in Wireless Sensor Networks

2.1 Basic Concepts

A join of two tables L and R consists of the concatenation of all the tuples from table L with those of table R where a condition (join predicate) is met on some attributes of the two tuples.

When an arbitrary comparison operator ($\geq$, $<$, $=$, ..) is allowed in join predicate, the join operation is called theta-join. An equijoin is a theta-join using the equality operator only.

In a wireless sensor network, a join is generally made between two regions of the sensor network. A region consists of a set of sensors of a geographically limited area.

2.2 Joins Implementation in Wireless Sensor Networks

A join operation in the wireless sensors networks can be accomplished according to two different implementations:

- **External join**

It is the simplest implementation of join operations in wireless sensor networks. It performs the join operation in the base station (sink), after transferring the tuples corresponding to the selection conditions.

This implementation consumes a lot of energy, because of the large amount of data to transmit.

- **In-network join**

Is the most complex implementation, but the more efficient. Join operation is performed by internal nodes of the network before transmitting final results to the base station.

2.3 Join types in Wireless Sensors Network

According to spatial or temporal aspect, the following types of joins operations are distinguished:

- According to spatial aspect.

- Inter-region joins.

They are the joins without spatial predicates; relations are easily identified by assuming that each node knows its location [Kang 13].

• **Unique region joins.**
They are joins with spatial predicates. A node receiving a query of this type, can't in general determine if it should participate to the query without communicating with others nodes. This may result in a difficult implementation comparing with an inter-region joins.

• According to the temporal aspect.

• **One execution joins.** A fixed window is specified for each of the two relations.

• **Continuous joins.** The relations use sliding windows or fixed windows defined for the future.

• **Periodic joins.** Based on jump windows, defining a time interval for repeating a query execution.

3. 'In-Network" Technique for N-Way Join Execution

3.1 Related Works

In-network techniques proposed for the join operations can be divided into two main categories: without filtering techniques, and with filtering techniques.

Several techniques that do not allow filtering of tuples before executing the join, have been proposed.

Yao and Gehrke [Yao 03] compared the transmission cost of an external join with that of a join computed in-network by an experiment. It was deduced that in-network processing allows less dissipation of energy for a low selectivity of joins.

Bonfils and Bonnet [Bonfils 03] studied the problem of searching the optimal node of an in-network join in the case that the operation it is carried out in only one node. The site of the optimal node considered is on the shortest path between the two nodes which produce detections, and more close to the node which produces more data. Madden and al. [Madden 03] treated a query join between a point of backup and a relation of sensor. Abadi and al. [Abadi 05] studied a join between an extern relation and the relation of the sensors.

Chowdhary and Gupta [Chowdhary 05] proposed technique for a continuous inter-region theta-join by separately considering the static cases of tables and data flow cases. In [Coman 07], Coman and al. presented the principle of the local join between two areas of a network of sensors. This technique consists in processing the join operation in one of the two areas after reception of the table from the other area. The result is then transmitted to the sink.

Pandit and Gupta [Pandit 06] proposed an algorithm based on an indexed and distributed data structure (distributed B+ tree) to process a continuous inter-region theta-join, with a predicate of interval. Pandit and Gupta [Pandit 06] also proposed 'Distributed Hash-Join' to treat continues inter-area theta join $R \bowtie S$, with interval predicates of the form: $|R.A - S.B|\ \theta$. The implementation adopted for this join is a hash-join algorithm.

Mihaylov and al. [Mihaylov 08] [Mihaylov 10] treated a continuous join with an expression more general, and modeled like join with a sliding window.

Concerning the techniques with filtering of tuples: Yu and al. [Yu 06] proposed Synopsis Join to process an immediate inter-region equi-join, by using a distributed alternative of the semi-join approach, so entirely reducing the tables of the join. Coman and al. [Coman 07] studied an inter-region equi-join of an immediate request. An alternative of semi-join is used to entirely reduce the two tables. The technique suggested is called: local semijoin. The join operation is carried out in one of the two areas.

Min and al [Min 11] proposed INJECT (In-Network Join strategy using Cost based optimization in Tree routing sensor networks) to treat a continuous theta-join on static relations. With INJECT, the authors conceived various plans of join in network environments of sensors.

Yang and al. [Yang 07] proposed the technique of auto-join (TPSJ) of two-phases to perform a continuous theta-join of the form $(\sigma p\ R) \bowtie_{A\theta B \wedge T(h)} R$..

CJF (Continuous Join Filtering) was proposed by Stern and al [Stern 10] to treat a continuous theta-join based on two static relations. It is a filtering approach that intends to keep filters and upgrade them. The reachable tuples are transmitted to the base station which carries out the final join.

Lay and al [Lai 08] proposed PEJA to answer a continuous inter-region equi-join. The algorithm uses histograms and subintervals of tuples to be able to filter the non-reachable subintervals before the transmission of the tuples to the base station for the final join.

Lay and al [Lai 10] also proposed SRJA to perform an inter-region equi-join, by carrying out a join of iceberg. The principle of a join of iceberg is to involve to the join result only the join attribute values of which the number of tuples jointed exceeds a given threshold α.

Stern and al. in [Stern 09] have proposed a solution allowing processing every type of joins in sensors networks (including n-way joins). This solution executes the joins at the base station (sink) in three steps:

- All join attributes are collected at the base station, to filter some values.
- The determined join filter is broadcasted in the network.
- Each site uses the filter to determine all the relevant tuples and send them to the sink, where the final join is processed.

This technique consumes too much energy because the number of tuples transmissions to the sink is generally very high.

3.2 Types of the Processed Joins

Are processed in our technique; the inter-region joins in one execution run, having syntax like this:

SELECT *R*1.*attrs*, *R*2.*attrs*, …, *Rn*.*attrs*
FROM *R*1, *R*2, …, *Rn*
WHERE *pred*(*R*1) AND *pred*(*R*2) … AND *pred*(*Rn*)
AND *join-exp* (*R*1.*join-attrs*, *R*2.*join-attrs* ,…, *Rn*.*join-attrs*)

With:
Ri is the relation of the i^{th} region.
pred (*Ri*) is a predicate of selection of relation *Ri*, and join-exp is the join condition.

An application example of these joins is the vehicle traffic control, through many geographical zones:

SELECT *Veh1.VehId*, *Veh*1.*time*, *Veh*2.*time*, *Veh*3.*time*
FROM *Veh*1, *Veh*2, *Veh*3
WHERE (*Veh*1.*time IN i*1) and (*Veh*2.*time in i*2) and (*Veh*3.*time in i*3) and (*Veh*1.*VehId* = *Veh*2. *Veh Id*) and (*Veh*2. *Veh Id* = *Veh*3. *Veh Id*)

Where: *i*1, *i*2, and *i*3 indicate time intervals during which the Vehicles passed respectively through areas 1, 2 and 3.

3.3 Technique Principle

N-way Local SemiJoin technique consists in:

- Extend the ideas developed for the binary joins in wireless sensor networks to the n-way joins.

- Adapt the technique of left linear trees to minimize the number of n-way joins [Steinbrunn 93].

- And consider geographical zone positions of participating zones to determine the execution order of joins (from the nearest to the farthest zone).

A join operation runs in three phases:

Phase 1 :
Transmission of the query from the sink to the root node of each area, using a location routing protocol GPSR [Karp 00]

[Ratnasamy 04], to ensure the arrival of the query message to the concerned regions. Each region consists of a tree of nodes. Each node must transmit its tuples to the root. The location of each node and the locations of its neighbors are assumed to be known via GPS or via localization algorithms [Savvides 04].

Phase 2 :
In-network execution of the join operation. The principle of the left linear trees is adopted to determine the execution order of joins. The joins are performed in pairs (Figure . 1). Where an unwinding in three steps for each pair of relation R*i* and Ri+1 performing a join operation:

Step 1: The join attribute of the relation R_{i+1} is transmitted to the zone of the relation $R_{i.}$

Step 2: Execution of semijoin in the region of R_i, and transmitting of the results to the region of relation R_{i+1}.

Step 3: Performing the join between the result of the semi-join and the relation of the region R_{i+1}.

Phase 3 :
Sending the final result of the area of last relation to the sink

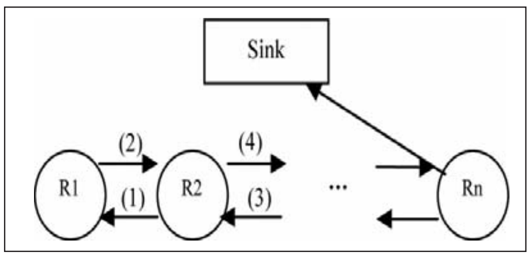

Figure 1. N-way local Semi-Join (NLSJ) execution

The principle of NLSJ will be illustrated with an example. For this purpose, a join between 3 tables is considered.

In a first step, the join is carried out between the relation R1 and the relation of the nearest area (R2). A projection of the R2 table on the join attribute is transmitted to the R1 area, where the semi-join with the table R1 will be carried out. The semi-join makes it possible to reduce the number of tuples in front of being transmitted to the R2 area. The result is transmitted to the R2 area, so that the end result of the join between R1 and R2 is given (Figure 2 (a)).

With the same manner, will be obtained, at the second step, the result of the join between the table R3 and the result of the join of R1 and R2 (Figure 2 (b)).

At the end, the result determined in the R3 area is sent to the sink (Figure 2 (c)).

4. Performance Analysis

4.1. Experimentation environment

To simulate n-way join executions, an NS3 simulator is used.

The technique has been applied on static tables of 2000 tuples each.

The message size is a tuple size which is 40 bytes each.

The column size is assumed to 10 bytes; the size of the result tuple is 30 bytes.

The communication cost was tested considering selectivity factors of intermediates joins in the interval [10^{-5}, 10^{-4}], and then in the interval [10^{-4}, 10^{-3}].

Selectivity factors of intermediates joins are generated in random way.

The values of the horizontal axis of results graphs are averages of selectivity factors of the intermediates joins.

Two simulation cases have been processed separately.

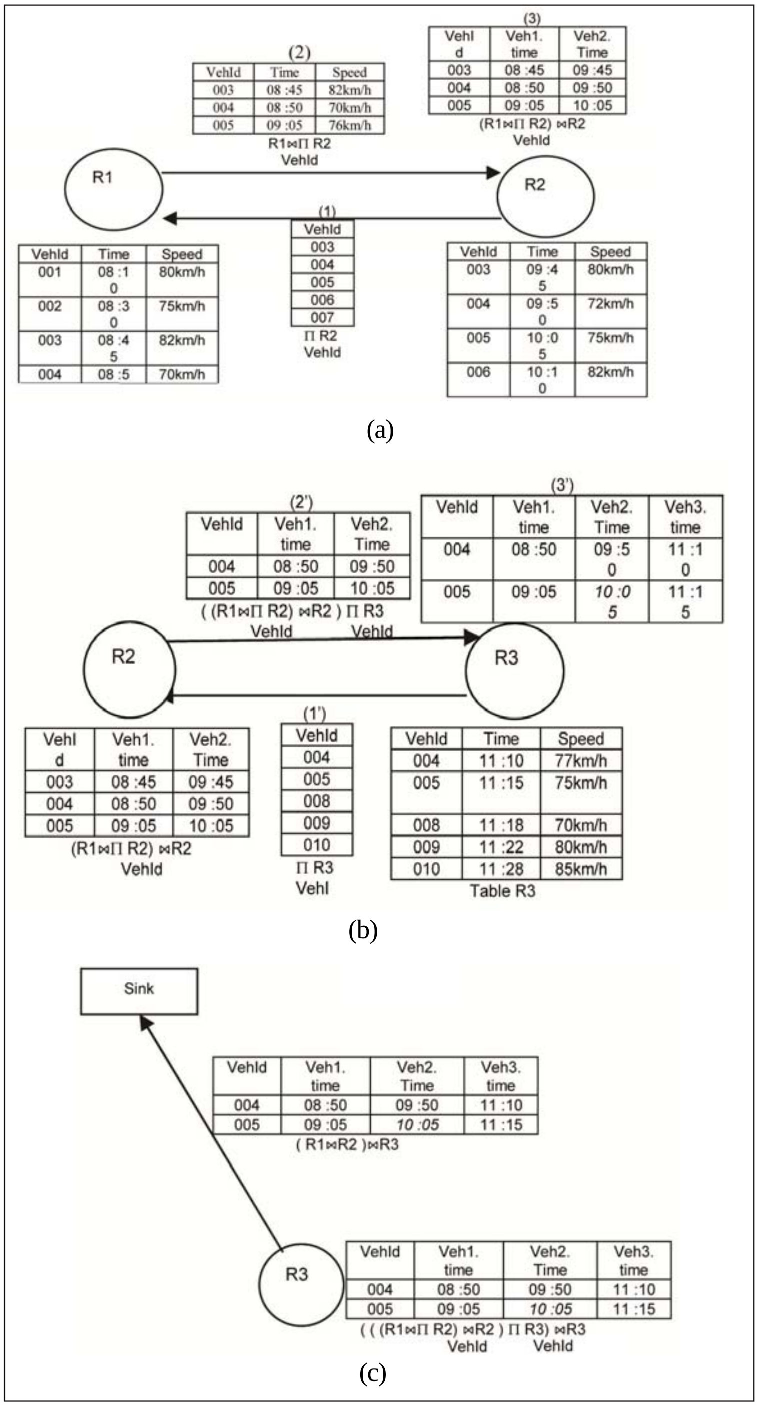

Figure 2. An example for N-way local Semi-Join (NLSJ) execution

- 3 tables simulation.

- 5 tables simulation.

For each simulation, 2 types of simulations have been performed.

- External join simulation: performing the join operation at the source.

- And the n-way local semi-join simulation, described above.

4.2 Experimentation Results

In the interval [10^{-5}, 10^{-4}] of selectivity factors, (figure 3 and figure 4), 'N-way Local Semi-Join' technique perform better than external join in both simulations: 3 tables and 5 tables.

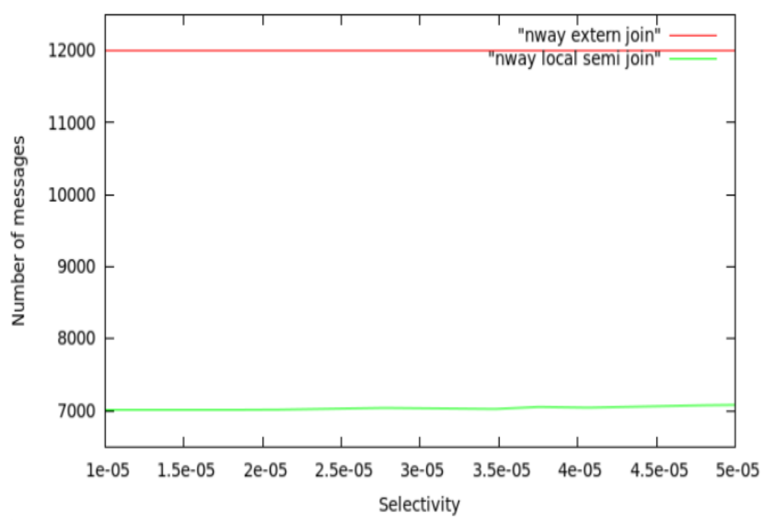

Figure 3. Communication cost for 3 tables following the average of selectivity factors of intermediates joins in the interval [10^{-5}, 10^{-4}]

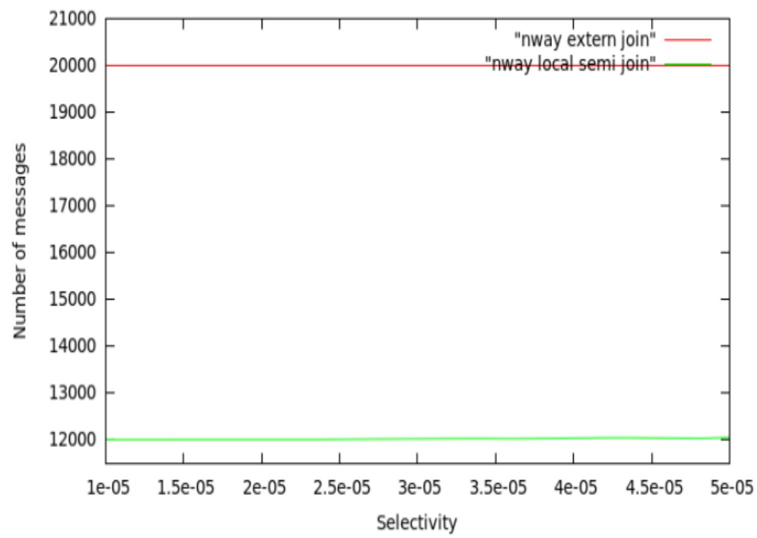

Figure 4. Communication cost for 5 tables following the average of selectivity factors of intermediates joins in the interval [10^{-5}, 10^{-4}]

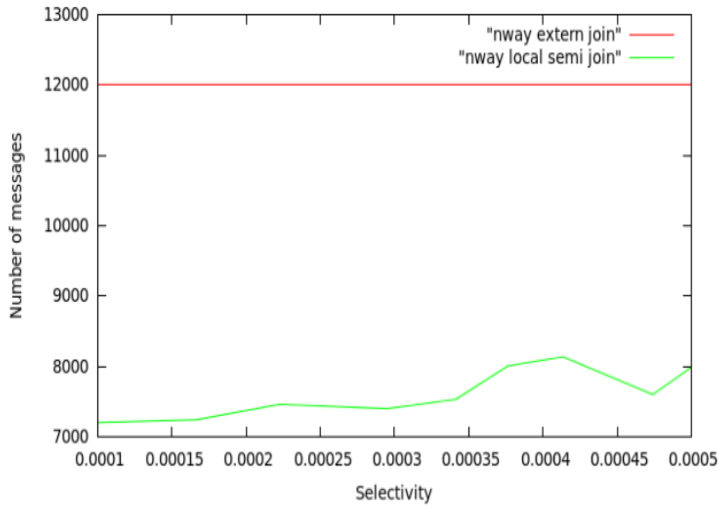

Figure 5. Communication cost for 3 tables following the average of selectivity factors of intermediates joins in the interval $[10^{-4}, 10^{-3}]$

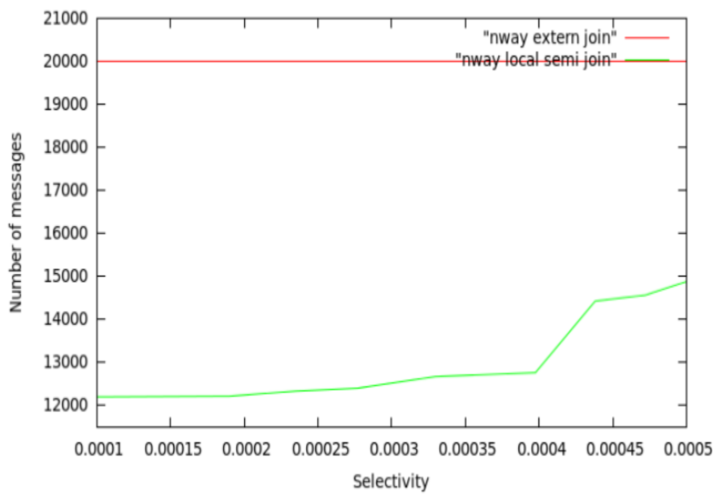

Figure 6. Communication cost for 5 tables following the average of selectivity factors of intermediates joins in the interval $[10^{-4}, 10^{-3}]$

In the interval $[10^{-4}, 10^{-3}]$ of selectivity factors, (figure 5 and figure 6), the 'N-way local semi-join technique' continue to offer the best performance.

With higher values of selectivity factors, the extern join is more efficient. Consumption in this case remains constant while it rapidly increases with the proposed technique.

5. Conclusion

In this paper, a technique for N-way join execution, in the wireless sensor network, is presented.

Adoption of semi-join technique principle and in-network execution principle, for the N-way joins, show a better performance compared with an execution at the source.

In future work, a study of joins operations between data streams composed by the sensor network tables is forecasted. Techniques suggested in the case for distributed databases were proposed in [Tri 10] and can be adapted to the cases of the databases of the networks of sensors.

References

[1] Abadi, D., Madden, S., Lindner, W. (2005). REED: Robust, Efficient Filtering and Event Detection in Sensor Networks. *In* : Proceedings of the 31*st* International Conference on Very Large Data Bases, Trondheim, Norway, 30 August–2 September; 769–780.

[2] Bonfils, B., Bonnet, P. (2003). Adaptive and Decentralized Operator Placement for In-Network Query Processing. *In*: Proceedings of International Workshop Information Processing in Sensor Networks, Palo Alto, CA, USA, 22–23 April, 47–62.

[3] Chowdhary, V., Gupta, H. (2005). Communication-Efficient Implementation of Join in Sensor Networks. *In:* Proceedings of 10th International Conference on Database Systems for Advanced Applications, Beijing, China, 17–20 April, 447–460.

[4] Coman, A., Nascimento, M., Sander, J. (2007). On Join Location in Sensor Networks. *In* : Proceedings of 8*th* International Conference on Mobile Data Management, Mannheim, Germany, 7–11 May, 190–197.

[5] Djail, B., Hidouci W-K. (2013). Les jointures dans les réseaux de capteurs sans fil. Conférence Nationale sur les Technologies de l'Information et les Télécommunications CNTIT'13, 10-11 Décember.

[6] Djail B., Hidouci, W-K. (2015). Les jointures des flux de données dans les réseaux de capteurs sans fil. 2ème Conférence Internationale sur les Nouvelles Technologies et la Télécommunication ICNTC'2015, 3-4 Mars.

[7] Hyunchul Kang. (2013). In-Network Processing of Joins in Wireless Sensor Networks http:// www . mdpi.com/journal/ sensors. 11 March .

[8] Karp, B., Kung, H. (2000). GPSR, Greedy Perimeter Stateless Routing for Wireless Networks. *In*: Proceedings of 6*th* Annual International Conference on Mobile Computing and Networking, Boston, MA, USA, 6–11 August, 243–254.

[9] Lai, Y., Chen, Y., Chen, H. (2013). PEJA: Progressive energy-efficient join processing for sensor networks. *J. Comp. Sci. Technol*. 2008, 6, 957–972. *Sensors*, 13, 3391.

[10] Lai, Y., Lin, Z., Gao, X. (2010). SRJA: Iceberg Join Processing in Wireless Sensor Networks. *In*: Proceedings of International Workshop on Database Technology and Applications, Wuhan, Hubei, China, 27–28 November, 1–4.

[11] Madden, S., Franklin, M., Hellerstein, J., Hong, W. (2003). The Design of An Acquisitional Query Processor for Sensor Networks. *In*: Proceedings of ACM SIGMOD International Conference on Management of Data, San Diego, CA, USA, 9–12 June, 491–502.

[12] Mihaylov, S., Jacob, M., Ives, Z., Guha, S. (2008). A Substrate for In-Network Sensor Data Integration. *In*: Proceedings of 5*th* Workshop on Data Management for Sensor Networks, Auckland, New Zealand, 24 August, 35–41.

[13] Mihaylov, S., Jacob, M., Ives, Z., Guha, S. (2010). Dynamic Join Optimization in Multi-Hop Wireless Sensor Networks. In: Proceedings of 36*th* International Conference on Very Large Data Bases, Singapore, 13–17, September, 1279–1290. Sensors 2013, 13,3392

[14] Min, J., Yang, H., Chung, C. (2011). Cost based in-network join strategy in tree routing sensor networks. *Inf. Sci.* 2011, 16, 3443–3458.

[15] Pandit, A., Gupta, H. (2006). Communication-Efficient Implementation of Range-Join in Sensor Networks. *In* : Proceedings of 11*th* International Conference on Database Systems for Advanced Applications, Singapore, 12–15 April, 859–869.

[16] Ratnasamy, S., Karp, B., Li, Y., Yu, F., Estrin, D., Govindan, R., Shenker, S. (2002). GHT: A geographic hash table for data-centric storage. *In*: Proceedings of WSNA'03.

[17] Savvides, A., Srivastava, M., Girod, Estrin, D. (2004). Localization in sensor networks. Wireless sensor networks, 327-349.

[18] Steinbrunn, M, Moerkotte, G., Kemper, A. (1993). Optimizing join orders. Technical report MIP- 9307, UniversitKt Passau, 94030 Passau, Germany.

[19] Stern, M., Buchmann, E., Böhm, K. (2009). Towards Efficient Processing of General-Purpose Joins in Sensor Networks. *In*: Proceedings of 25*th* IEEE International Conference on Data Engineering, Shanghai, China, 29 March–2 April, 126–137.

[20] Stern, M., Böhm, K., Buchmann, E. (2010). Processing Continuous Join Queries in Sensor Networks: A Filtering Approach. *In:* Proceedings of ACM SIGMOD International Conference on Management of Data, Indianapolis, IN, USA, 6–11 June, 267–278.

[21] Tran, Minh. (2001). Byung Suk Lee. Distributed stream join query processing with semijoins. Springer Science+Business Media, 6 March, 211–254.

[22] Yang, X., Lim, H., Özsu, M., Tan, K. (2007). In-Network Execution of Monitoring Queries in Sensor Networks, *In*: Proceedings of ACM SIGMOD International Conference on Management of Data, Beijing, China, 12–14 June, 521–532.

[23] Yong Yao, Y., Gehrke, J. (2003). Query Processing for Sensor Networks. *In:* Proceedings of 1st Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, 5–8 January.

[24] Yu, H., Lim, E., Zhang, J. (2006). On In-Network Synopsis Join Processing for Sensor Networks. *In:* Proceedings of 7th International Conference on Mobile Data Management, Nara, Japan, 9–13 May, 32–39.

[25] Zhao, F., Guibas, L. (2004). Wireless Sensor Networks: An Information Processing Approach. Morgan Kaufmann: San Francisco, CA, USA.