Soumis le : 14/08/2016
Forme révisée acceptée le : 30/10/2017
Auteur correspondant : b_djail@esi.dz

Nature & Technology

http://www.univ-chlef.dz/revuenatec

An energy-efficiency technique for n-way stream joins in wireless sensor networks

Djail Boubekeur*, Hidouci Walid Khaled and Loudini Malik

LCSI Laboratory National High School for Computer Science (ESI) Algiers

Abstract

In wireless sensor networks, the critical factor is the sensors energy. The exhaustion of this energy causes network paralysis. In wireless sensor networks, the join operation needs a lot of energy consumption. This energy is so very high for nway join queries, and more very high for nway stream queries. Little work has been done to address this problem. We present in this paper, an energy-efficiency technique for n-way joins execution between data streams in wireless sensor networks. The experiment conducted has shown good performance of the proposed technique compared to SENS-join technique.

Keywords : wireless sensor networks; communication cost; in-network join; n-way stream join

1. Introduction

A wireless sensor network is characterized by the low cost and low power (calculation and storage) sensor component. The sensors communicate via a wireless medium and work together to accomplish a common goal. Each sensor generates records following the various collection actions. These tuples constitute a local data table to the sensor.

All local tables' nodes of the same type of the same sensor network form a distributed table [1]. We use traditional relational query for questioning the database of the wireless sensor network. Joins queries are widely used in wireless sensor networks' applications that require data gathering from multiple nodes. The joins can be performed between static tables, as they can run between data streams. Join queries and particularly joins between data streams have a very high energy consumption of the sensors, which could lead to the disappearance of the network.

In wireless sensors network, data transmission consumes more energy than data processing in the node [2]. It is important to reduce the quantity of transmitted data during a join operation in order to decrease the energy consumed by the sensors. The main challenge then is to minimize the energy consumed. This being our primary goal in this paper, and we propose an in-network technique for join queries between data streams in sensor networks. The techniques proposed recently such as Distributed-Broadcast of Coman and al. [3], Mediated Join of Pandit and Gupta [4], Synopsis Join of Yu and al [5], and INJECT of Min and al [6], have been limited to binary joins queries. N-way joins (joins between more than two tables or streams) are rarely addressed.

This paper presents an energy-efficiency technique for n-way joins execution in wireless sensor networks. The rest of the paper is organized as follows: section 2 provides a description of the general characteristics of joins operations in wireless sensor networks. Section 3 gives related work. Section 4 describes our proposed technique. Section 5 contains the results of the simulation performed, and a discussion of the obtained results. Finally, section 6 concludes the paper.

2. General characteristics of joins 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 (≥, <, =, ...) is allowed in join predicate, the join operation is called theta-join. An equijoin is a theta-join using only the equality operator. In a wireless sensor network, a join is generally made between two regions of the sensors network. A region consists of a set of sensors of a geographically limited area.

2.2. Joins implementations in wireless sensor networks.

There are two join implementations in wireless sensor networks:

- The first one is the external join. It executes the join operation at the base station after transferring the tuples from the nodes. This implementation consumes a lot of energy because of the high amount of transmitted data.
- The second one is the 'In-network' implementation, which performs the join operation at the nodes before transmitting the result to the sink. This is the most efficient implementation.

2.3. Join types in the wireless sensors network.

According to spatial aspect, we distinguish an inter-region join and a unique region join. The inter-join operation is running between two tables from two regions of the sensor network. The unique region join performs the join between the sensors tuples located in the same region. A node receiving a query of this type, can't in general, determine if it should participate in the query without communicating with other nodes. This type of join is very difficult to implement.

According to the temporal aspect, we can class the joins operations in wireless sensors networks into three categories:

- One execution joins. A fixed window is specified for each of each two relations.
- Continuous joins. They use sliding windows or fixed windows defined for the future.
- Periodic joins. They are based on jump windows, defining a time interval for repeating query execution.

3. Related works

Several techniques were proposed to execute join operations in wireless sensors network. They can be divided into two main categories: without filtering and with filtering techniques. Techniques without filtering execute join operations between tuples of tables or of data streams. This causes a great consumption of energy, due to the high number of transmitted messages. It was the first proposed techniques [1, 4, 7-12].

Techniques with filetring use the semi join principle to eliminate non joingnables tuples before the join execution. The objective of the filtering techniques is to minimize the number of transmitted messages. These techniques were proposed in several works [3, 5, 6, 13-16]. All those techniques were suggested for the binary join. Few those developed for nway join. Stern and al. in [17] have suggested a solution allowing processing every joins type 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 broadcast 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 transmitted to the sink is very high.

4. N-WAY STREAM LOCAL JOIN description

We propose N-way Stream Local Join (NSLJ) to reduce efficiently the consumed energy for n-way joins between data stream. We process a continuous inter-region join having syntax like this:

```
SELECT S1.attrs, S2.attrs,…,Sn.attrs
FROM  S1, S2, …, Sn
WHERE  pred(S1)  AND pred(S2)  … AND pred(Sn)
AND join-exp (S1.join-attrs , S2.join-attrs ,…, Sn.join-attrs)
```

where: pred (Si) a predicate of selection of steam Si, and join-exp the join condition.

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

```
SELECT    Veh1.VehId, Veh1.time, Veh2.time, Veh3.time
FROM      Veh1, Veh2, Veh3
WHERE        (Veh1.time  IN  i1)  and  (Veh2.time  in  i2)  and
(Veh3.time in i3) and (Veh1.VehId = Veh2. Veh Id) and (Veh2. Veh
Id= Veh3. Veh Id)
```

where: i1, i2, and i3 indicate time intervals during which vehicles passed respectively through regions 1, 2 and 3.

N-way Stream Local Join (NSLJ) consists of executing each join operation at a local node in the network. We also adapt the technique of left linear trees to minimize the number of n-way joins [18], and we consider geographical zone positions of participating zones to determine the execution order of joins (from the nearest to the farthest zone). In addition, we accommodate the technique 'distributed one-way stream join's processed at destination proposed in [19]. This technique executes a join operation between stream data, with considering tuples' windows at each operation.

By NSLJ, a join operation runs in two phases:

Phase 1:

The query is transmitted from the sink to the root node of each area, using a location routing protocol GPSR [20] [21], 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. We assume that each node knows its location and the locations of its neighbours, via GPS or via localization algorithms [22].

Phase 2:

A set of tuples B11 are transmitted from the buffer of S1 to the region of S2. In this region, is executed the join between B1 and the windows W22 maintained at S2. The result of the join is forwarded to the region of S3.

The process is repeated between each stream Si and the following stream Si+1 until determining the end result and transmitting it to the sink (Figure 1).

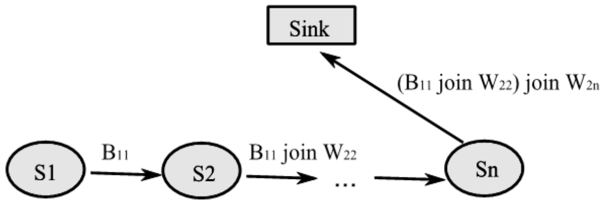

Figure 1: Nway Stream local Join (NSLJ) execution.

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

In a first step, the join is performed between a set of tuples B11 of the stream S1 and a window of the stream of the nearest region (S2). So, B11 is transmitted to the S2 region, then the result of the join transmitted to the S3 region (Figure 2 (a)). With the same way, will be obtained, at the second step, the result of the join between S3 and the result of the join of B11 and a window of S2 (Figure 2 (b)). And at the last step, the result determined in the S3 region is sent to the sink (Figure 2 (b)).

To cite this article: Djail Boubekeur, Hidouci Walid Khaled and Loudini Malik, An energy-efficiency technique for n-way stream joins in wireless sensor networks, Nature & Technology Journal. Vol. A: Fundamental et Engineering Sciences, 18 (2018) 49-55
http://www.univ-chlef.dz/revuenatec/issue-18/Article_A/Article_452.pdf

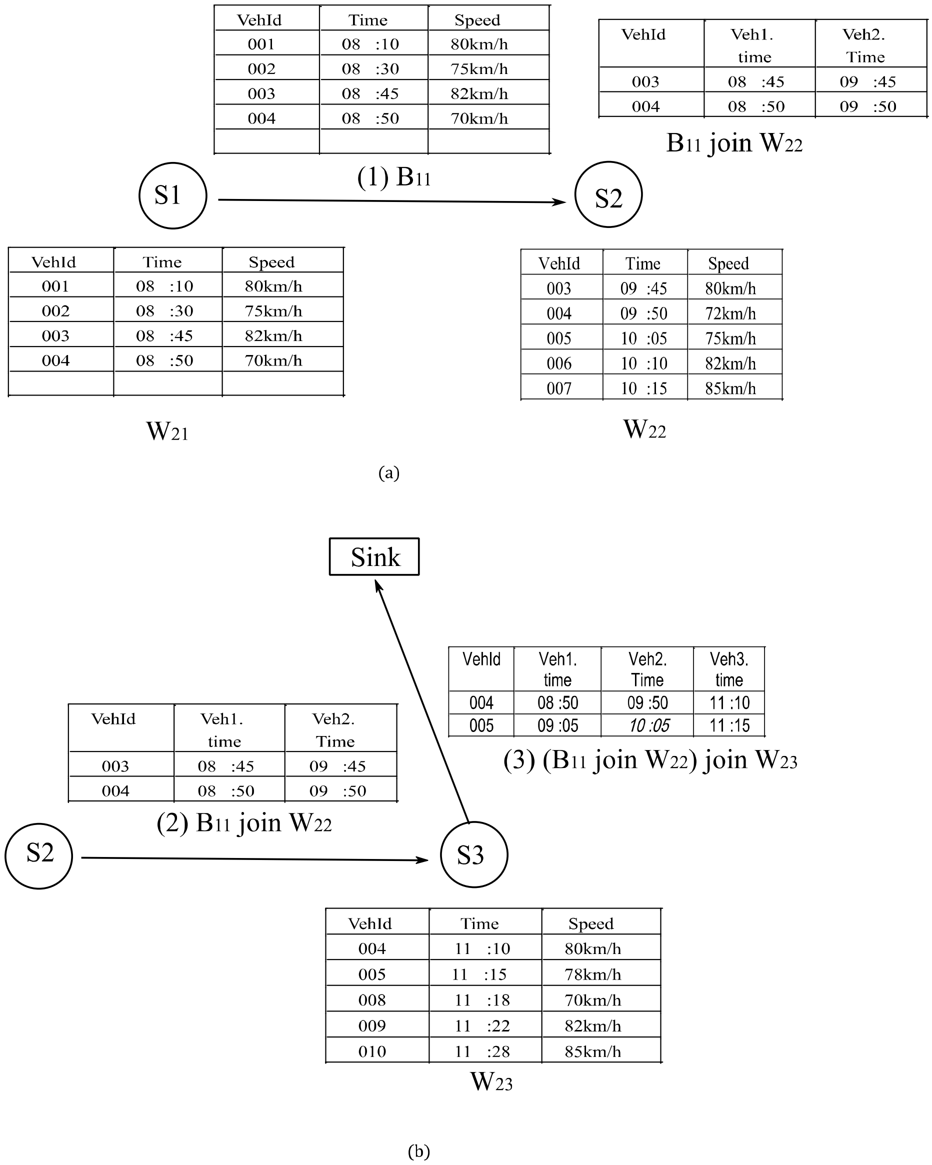

Figure 2 - An example for N-way Stream local Join (NSLJ) execution.

5. Performance analysis

5.1. Experimentation environment

To simulate the n-way join execution, we used the NS3 simulator. The technique has been applied considering a message size equals a tuple size, which is equal 40 bytes. The column size is supposed 10 bytes; the result tuple size is 30 bytes. The communication cost was tested considering intermediates joins' selectivity factors in the intervals $[10^{-5}, 10^{-4}]$ and $[10^{-4}, 10^{-3}]$. Intermediates joins' selectivity factors are generated in random way. The values of the horizontal axis of results graphs represent averages of the intermediates joins selectivity factors.

We process separately two simulations: 3 streams simulation and 5 streams simulation. For each simulation, we perform 2 cases: The SENS-join simulation and our technique (NSLJ) simulation.

5.2. Experimentation results

In the interval of $[10^{-5}, 10^{-4}]$ of selectivity factors, and with 5 streams (Figure 4), N-way Stream Local Join performs better than SENS-join. In the same interval (Figure 3) and with 3 streams our technique drops in performance. In the interval $[10^{-4}, 10^{-3}]$ of selectivity factors (Figure 5 and Figure 6) 'N-way Stream local join' present lost results.

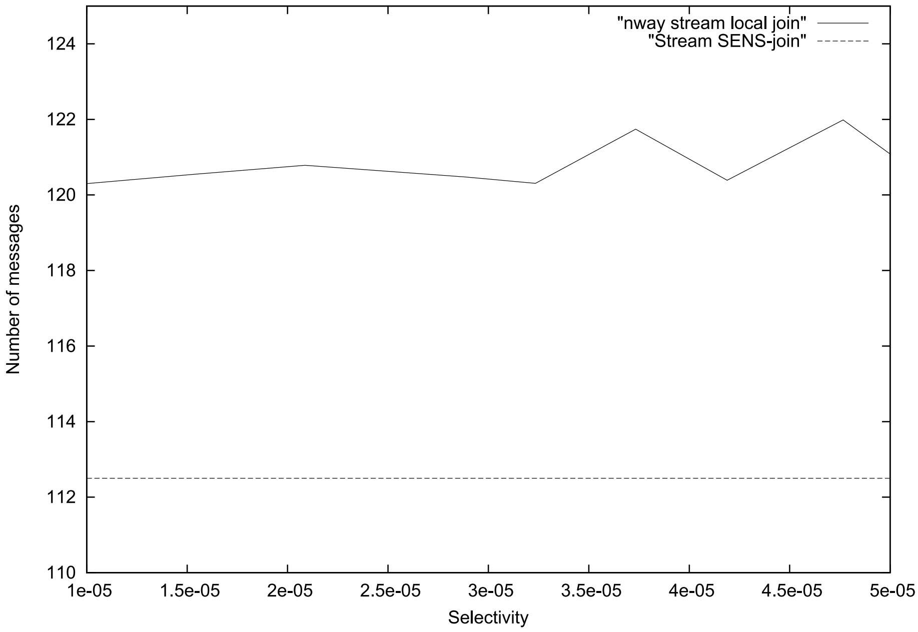

Figure 3: Communication cost for 3 streams following the average of intermediates joins' selectivity factors in the interval [10^{-5}, 10^{-4}]

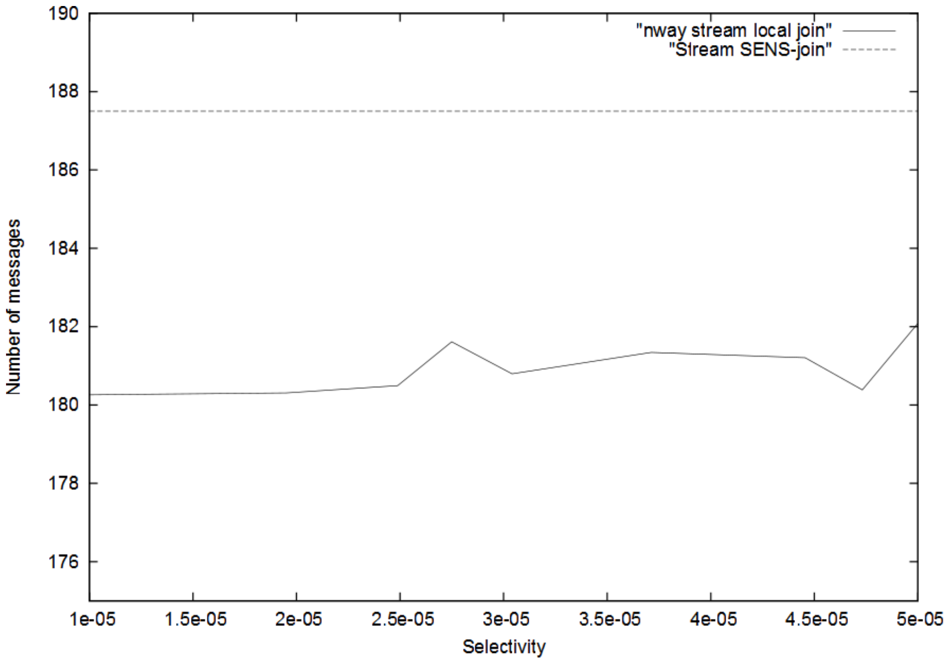

Figure 4: Communication cost for 5 streams following the average of intermediates joins' selectivity factors in the interval [10^{-5}, 10^{-4}]

To cite this article: Djail Boubekeur, Hidouci Walid Khaled and Loudini Malik, An energy-efficiency technique for n-way stream joins in wireless sensor networks, Nature & Technology Journal. Vol. A: Fundamental et Engineering Sciences, 18 (2018) 49-55
http://www.univ-chlef.dz/revuenatec/issue-18/Article_A/Article_452.pdf

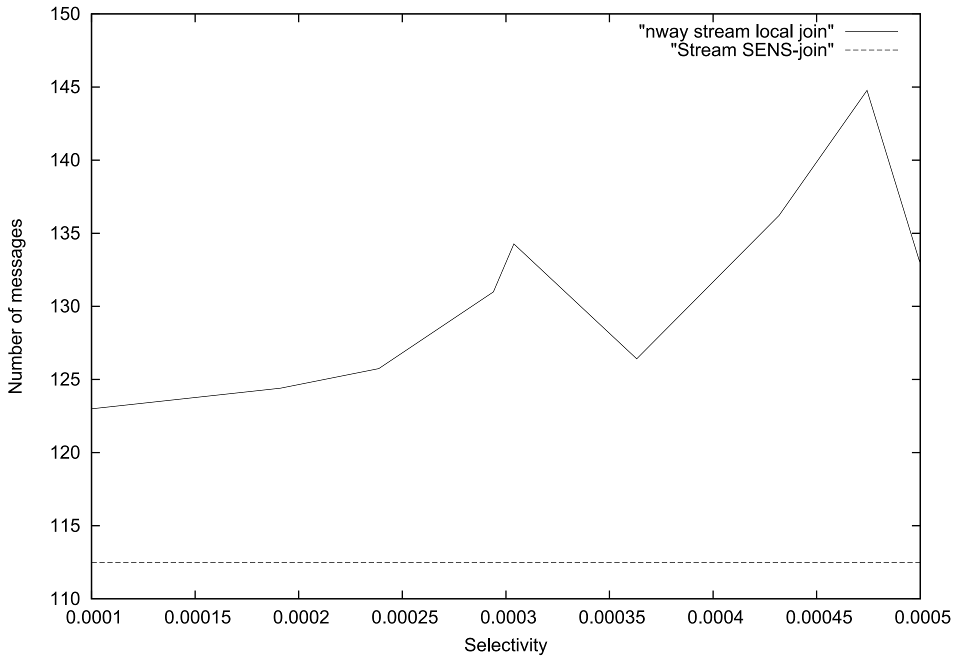

Figure 5: Communication cost for 3 streams following the average of intermediates joins' selectivity factors in the interval [10^{-4}, 10^{-3}]

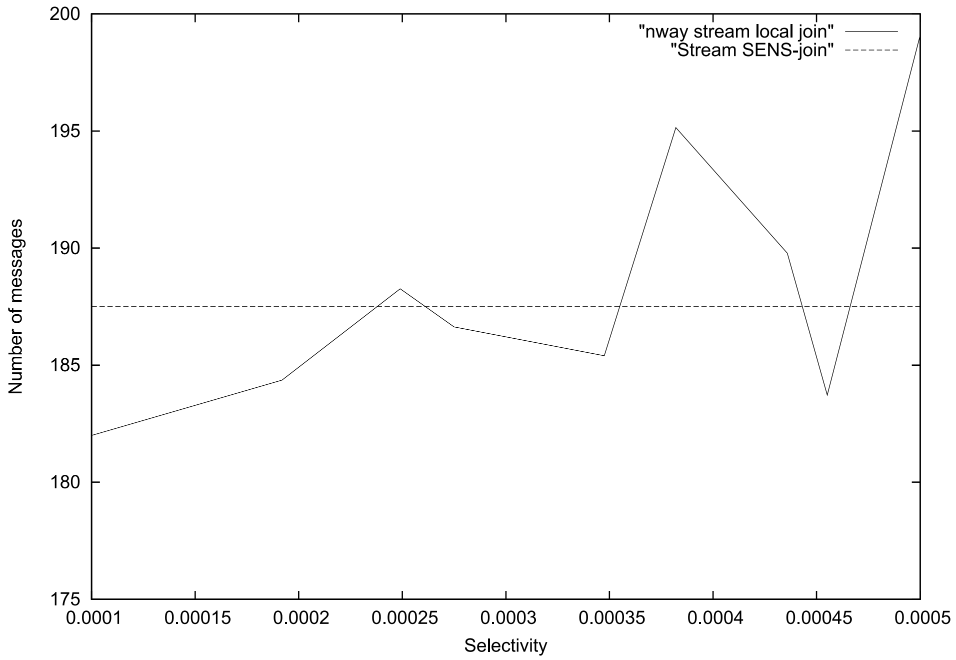

Figure 6: Communication cost for 5 streams following the average of intermediates joins' selectivity factors in the interval [10^{-4}, 10^{-3}]

5.3. Discussion

With NLSJ, we execute the join query at the nodes. The join result volume decrease after each operation performed. NLSJ performs better for a high number of streams, and a very low selectivity factor. While SENS-join technique has a constant performance level, because the information volume transmitted to the sink is constant for each case. It is important to note that these results are valid for low selectivity values. For high selectivity values, it is ideal to realize the join query at the sink.

6. Conclusion

We have presented, in this paper, an energy-efficiency technique for n-way stream join execution, in wireless sensor networks. We have adopted the principle of in-network execution to reduce the number of tuples transmitted between sensors and to sink.

Our approach shows better performance compared to SENS-join principle for low selectivity factor and high number of streams.

In future work, we plan to adapt the semi-join principle in order to reduce effectively the number of transmitted messages and then the consumed energy.

References

[1] Yao Y. and Gehrke J., "Query Processing in Sensor Networks," In CIDR , 2003, pp. 233-244.

[2] Zhao F. and Guibas L. J., Wireless sensor networks: an information processing approach: Morgan Kaufmann , 2004.

[3] Coman A., Nascimento M. A., and, J. Sander , "On join location in sensor networks," In The 8th International Conference on Mobile Data Management, 2007, pp. 190-197.

[4] Pandit A. and Gupta H., "Communication-efficient implementation of range-joins in sensor networks, In "The 11th International Conference on Database Systems for Advanced Applications", 2006, pp. 859-869.

[5] Yu H., Lim E.-P., and Zhang J., In "On in-network synopsis join processing for sensor networks", In 7th International Conference on Mobile Data Management (MD M'06), 2006, pp. 32-32.

[6] Min J.-K., Yang H., and Chung C.-W., In "Cost based in-network join strategy in tree routing sensor networks", Information Sciences, vol. 181, pp. 3443-3458, 2011.

[7] Bonfils B. J. and Bonnet P., In "Adaptive and decentralized operator placement for in-network query processing", Telecommunication Systems, vol. 26, pp. 389-409, 2004.

[8] Madden S., Franklin M. J., Hellerstein J. M., and Hong W., In "The design of an acquisitional query processor for sensor networks", In Proceedings of the 2003 ACM SIGMOD international conference on Management of data, 2003, pp. 491-502.

[9] Abadi D. J., Madden S., and Lindner W., "*Reed: Robust, efficient filtering and event detection in sensor networks*", In Proceedings of the 31st international conference on Very large data bases, 2005, pp. 769-780.

[10] Chowdhary V. and Gupta H., "*Communication-efficient implementation of join in sensor networks*", Iin International Conference on Database Systems for Advanced Applications, 2005, pp. 447-460.

[11] Mihaylov S. R., Jacob M., Ives Z. G., and Guha S., "*A substrate for in-network sensor data integration*", In Proceedings of the 5th workshop on Data management for sensor networks, 2008, pp. 35-41.

[12] Mihaylov S. R., Jacob M., Ives Z. G., and Guha S., "*Dynamic join optimization in multi-hop wireless sensor networks*", Proceedings of the VLDB Endowment, vol. 3, pp. 1279-1290, 2010.

[13] Yang X., Lim H. B., Özsu T. M., and Tan K. L., "*Network execution of monitoring queries in sensor networks*", In Proceedings of the 2007 ACM SIGMOD international conference on Management of data, 2007, pp. 521-532.

[14] Stern M., Böhm K., and E. Buchmann, "*Processing continuous join queries in sensor networks: a filtering approach*", In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, 2010, pp. 267-278.

[15] Lai Y. X., Chen Y. L., and Chen H.", *PEJA: Progressive energy-efficient join processing for sensor networks*", Journal of Computer Science and Technology, vol. 23, pp. 957-972, 2008.

[16] Lai Y., Lin Z., and Gao X., "*SRJA: Iceberg Join Processing in Wireless Sensor Networks*", In 2010 2nd International Workshop on Database Technology and Applications, 2010, pp. 1-4.

[17] Stern M., Buchmann E., and Böhm K., "*Towards efficient processing of general-purpose joins in sensor networks*", In 2009 IEEE 25th International Conference on Data Engineering, 2009, pp. 126-137.

[18] Steinbrunn M., Moerkotte G., and Kemper A., Optimizing join orders: Citeseer, 1993.

[19] Tran T. M. and Lee B. S., "*Distributed stream join query processing with semijoins*", Distributed and Parallel Databases, vol. 27, pp. 211-254, 2010.

[20] Karp B. and Kung H.-T., "*GPSR: Greedy perimeter stateless routing for wireless networks*", In Proceedings of the 6th annual international conference on Mobile computing and networking, 2000, pp. 243-254.

[21] Ratnasamy S., Karp B., Yin L., Yu F., Estrin D., Govindan R., and Shenker S., "*GHT: a geographic hash table for data-centric storage*", In Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, 2002, pp. 78-87

[22] Savvides A., Srivastava M., Girod L., and Estrin D., "*Localization in sensor networks*", In Wireless sensor networks, ed: Springer, 2004, pp. 327-349.

To cite this article: Djail Boubekeur, Hidouci Walid Khaled and Loudini Malik, An energy-efficiency technique for n-way stream joins in wireless sensor networks, Nature & Technology Journal. Vol. A: Fundamental et Engineering Sciences, 18 (2018) 49-55
http://www.univ-chlef.dz/revuenatec/issue-18/Article_A/Article_452.pdf