



Comparative Evaluation of Techniques for n-way Stream Joins in Wireless Sensor Networks

Djail Boubekeur

LCSI Department, High School of Computer Science (ESI), Algiers 16000
b_djail@esi.dz

KEYWORDS

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

ABSTRACT

In wireless sensor networks, sensor data are accessed using relational queries. Join queries are commonly used to retrieve the data from multiple tables stored in different parts of a wireless sensor network. However, such queries require large amounts of energy. Many studies have intended to reduce query energy consumption. However, most of the proposed techniques addressed binary joins which are performed between static tables. N-way joins between data streams were rarely considered. Join queries using data streams work continuously and require increasing energy, which is why n-way joins involving several tables consume so much energy. Thus, the challenge lies in reducing energy dissipation. Additionally, it is necessary to determine the appropriate execution order for an n-way join. The number of possible implementations of an n-way join grows exponentially with the tables' number. In this paper, interesting approaches for n-way joins between streams of data are evaluated. The methods that have been compared are extern-join, Sens-join of Stern et al, and the two techniques NSLJ (N-way Stream Local Join) and NSLSJ (N-way Stream Local Semi-Join). Comparisons are conducted according to several parameters to determine which use case is appropriate for each technique. NSLSJ works best for join queries with low join selectivity factors, while extern-join is more suitable for queries with very high selectivity factors.

1. Introduction

A wireless sensor network regroups a set of nodes, where each node corresponds to a sensor. Sensors have limited memory and computing capabilities.

Sensors detect events and save corresponding data. Data records at each node form a dataset. The data of all nodes make up a distributed database table. Access to data is performed by relational queries. Joins are relational queries widely used in wireless sensor networks, such as in the following applications: vehicle surveillance and tracking, animal habitat monitoring, environment monitoring, home and commercial building automation, precision agriculture, and water resource management (Kang, 2013).

A join query consists of assembling data from many nodes of the same network. This query type requires higher energy consumption. Since the nodes have limited energy, join execution might cause the failure of each node and the entire network. With n-way join queries in wireless sensor networks, the challenge lies in significantly reducing energy consumption. Additionally, it is necessary to determine the most appropriate order for the execution of the query. Note that the number of order possibilities grows exponentially with the number of tables being considered.

Several techniques have been proposed to treat binary joins and joins between static tables. Established solutions consist in reducing the number of messages that are sent. Thus, it was confirmed that energy consumption is higher during the transmission of messages than it is during data processing at the nodes. For continuous n-way joins, several studies have been carried out, such as SENS-join (Stern, Buchmann, and Böhm, 2009) of Stern et al., NSLJ (N-way Stream Local Join) (Djail, Hidouci, and Loudini, 2016) and NSLSJ (N-way Stream Local Semi-Join)(Djail, Hidouci, and Loudini, 2019). SENS-join determines a filter at the sink, then communicates it to interval nodes to recuperate joinable tuples. The final result is calculated at the base station. NSLJ performs the join query at internal nodes, without using filters. NSLSJ improves NSLJ by adopting semi-join to filter the non-joinable tuples.

The remainder of the paper is organized as follows: Section 2 summarizes the main features of join queries in wireless sensor networks. Section 3 describes related work. Section 4 presents the principles of the techniques selected for comparison. Section 5 discusses the conducted tests and the obtained results. Finally, the last section concludes the paper.

2. Join Features in Wireless Sensor Networks

2.1. Definitions

A join between two tables R and S is the table that contains tuples of R matched tuples of S according to a condition called: join predicate. When the join is performed between two tables, it is referred to as a binary join. If more than two tables are considered, the join is an n-way join.

A join with join predicates that use arbitrary comparison operators defines a theta-join. An equi-join uses only the equality operator in its join predicate.

2.2. Implementation of Join Queries in Wireless Sensor Networks

There are mainly two implementations of join queries in wireless sensor networks: extern and in-network join executions. (Kang, 2013)

Extern-join executes the join query entirely at the sink. The tuples of designed tables must be sent to the station base. It is the easiest to implement, but also the most energy-intensive.

In-network implementation decreases consumed energy considerably by reducing the number of transmitted messages by performing the query at the internal nodes (Kang, 2013).

2.3. Join Types in Wireless Sensors Network

Considering the spatial aspect, the joins in the wireless sensor networks can be divided into two large classes: unique-region joins and inter-region joins. (Kang, 2013)

A unique-region join is performed between nodes tuples of the same region, while, an inter-region join (Figure 1) is executed between nodes tuples of two distinct regions.

Considering the temporal aspect, join queries in wireless sensor networks can be split into two categories: One-shot joins and continuous joins.

With one-shot joins, a fixed window is defined for each table. It corresponds to a tuples number or to a time period. Query joins are performed on windows as a single execution using static tables.

With continuous joins, sliding windows are used to permit an ongoing execution of the query. A continuous join that is performed repeatedly at periodic intervals is called periodic join.

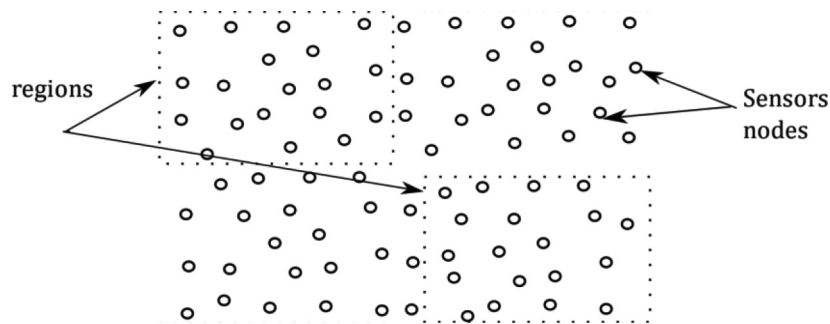


Figure 1. Inter-region join principle

3. Related Works

Join queries have been addressed by many studies. Thus, the proposed techniques can be divided into two large categories: techniques with filtering of non-joinable tuples and techniques without filtering.

Joins without filtering are performed by considering all the tuples of the tuples. Thus, no filter was generated or applied. Yao and Gehrke (Yao and Gehrke, 2003) compared an extern join to an in-network join in terms of communication costs. They concluded that the in-network technique permits less energy consumption for a low join selectivity. Bonfils and Bonnet (Bonfils and Bonnet, 2004) assessed the optimal node for an in-network join. The site situated on the shortest path between the two nodes participating in the query is the winner. Coman et al. (Coman, Nascimento, and Sander, 2007) presented local join and mediated join techniques for treating an inter-region join. Local join performs the query locally at nodes of one of the two regions. However, mediated join executes the query at an intermediate region. It has resulted in that no specific join strategy has the best performances for all queries.

The techniques with filtering adopt various principles, such as semi-join, to filter non-joinable tuples. These techniques permit a considerable gain of energy and are most recently used. Yu et al. (Yu, Lim, and Zhang, 2006) presented Synopsis Join to deal with a one-shot inter-region join. They adopt a distributed alternative of the semi-join approach to decrease tables' sizes. Coman et al. (Coman et al., 2007) proposed Local Semi-Join technique based on the semi-join principle. The join operation is executed in one of the two regions. Min et al (Min, Yang, and Chung, 2011) suggested various plans to perform a join query and they developed a cost model to select the optimal plan under various conditions.

Other authors addressed specific join queries. Mo et al. (Mo, Fan, Li, and Wang, 2014) treated spatial queries in a wireless sensor network. Kang et al (Kang, 2015) addressed the iceberg join query, a special type of join where only tuples whose cardinality exceeds a certain threshold are accepted to the join operation. Min et al. (Min, Kim, and Shim, 2014) presented a solution-based time-windowed principle to treat continuous joins.

These techniques were mostly proposed for binary joins. Few studies addressed n-way join and joins between data streams. Stern et al. in (Stern et al., 2009) suggested a strategy to treat all join types, including n-way join queries. The strategy consists in performing the join at the sink, by using filters that are determined at internal nodes based on the relevant records. Djail et al. proposed NLJ (N-way Local Join) (Djail, Hidouci, and Loudini, 2015), NLSJ (N-way Local Semi Join) (Djail et al., 2016) and NMSJ (N-way Mediated Semi Join) (Djail, Hidouci, and Loudini, 2018) (Djail, Hidouci, and Loudini, 2020) techniques to address n-way join between static tables and NSLJ (Boubekeur, Khaled, and Malik, 2018) and NSLSJ (Djail et al., 2019) to treat n-way join between data streams. NLSJ and NSLSJ apply semi-join to filter non-joinable tuples.

4. The Evaluated Techniques

This study evaluates four approaches to treating n-way joins between streams in wireless sensor networks. Among them are the extern join for data streams, the Sens-join developed by Stern et al., and the NSLJ and NSLSJ proposed by Djail et al. The evaluation is based on estimating the communication cost of each research technique. The communication cost corresponds to the number of transmitted messages. These four techniques are described here:

4.1. N-way Stream Extern Join Technique

Extern joins are used to join data streams at the base station. Periodically, different sites transmit sets of tuples to the sink. For each set reception, a join query is executed (Kang, 2013) (Figure 2).

The extern join technique is simple to use, but its transmission cost is extremely high.

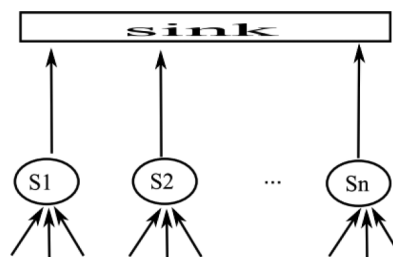


Figure 2. Extern join execution

4.2. N-way Stream Local Join Technique

N-way Stream Local Join (NSLJ) (Djail et al., 2019) treats n-way join queries between data streams in wireless sensor networks. NSLJ does not adopt tuples filtering. It executes intermediate joins for an n-way stream join at in-network by selecting the destination node to do this operation.

NSLJ uses the left linear trees technique to choose the best execution order of the join operations (Steinbrunn, Moerkotte, and Kemper, 1993). This choice is guided by knowing geographical zone positions to select the nearest region as the next destination node.

Additionally, NSLJ adopts the principle of the technique ‘distributed join processed at destination node’ proposed in (Tran and Lee, 2010) for classical distributed systems.

NSLJ technique runs in three phases:

Phase 1. Query dissemination

Initially, the query is first generated at the base station. Then, it is transmitted to the specified regions. A location routing protocol such GPSR (Karp and Kung, 2000) is used to ensure that the query is correctly received at root nodes of regions. A root node is a principal node at a region organized in tree, where it is assumed that each node knows its location and the locations of its neighbors, via GPS or via localization algorithms (Savvides, Srivastava, Girod, and Estrin, 2004).

Phase 2. Query execution

An n-way join executes several operations. At each operation, an intermediate join is performed. An operation is realized between a pair of nodes. Node pairs are determined on the basis of a technique of left linear trees which, at each step, select the next node to participate in the following operation.

Assuming a node pair (S_i, S_{i+1}) , NSLJ performs the following (Figure 3):

- A set of tuples (B_{i1}) is transmitted from a site S_i to a site S_{i+1} .
- An intermediate join is performed between B_{i1} and the window W_{i2} which is maintained at the site S_{i+1} .

The result of an intermediate join is communicated to the next node, which repeats the same steps until the final result is determined at the site

Phase 3. Final result transmission

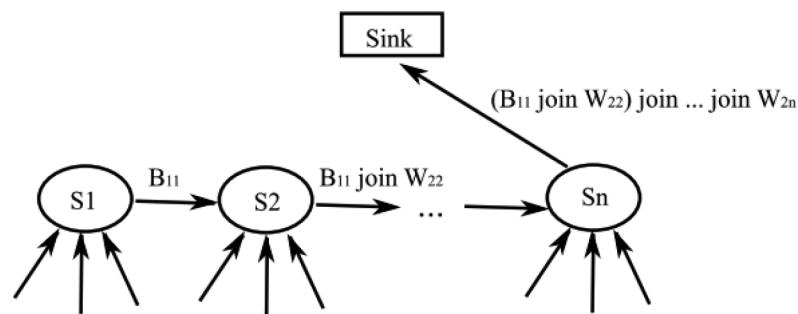


Figure 3. N-way Stream Local Join (NSLJ) execution

Once the intermediate joins have been completed, the final result is communicated to the base station.

4.3. N-way Stream Local Semi-join Technique

N-way Stream Local Semi-Join (NSLSJ) (Djail et al., 2019) is a filtering technique that uses the semi-join principle to filter tuples and improve the performances of the latest proposed technique NSLJ (Boubekeur et al., 2018) and NMSJ (Djail et al., 2018).

NSLSJ performs a join query in three phases:

Phase 1. Query dissemination

In the same way as the NSLJ technique described before, this phase is performed. The query is communicated to root nodes using a location protocol such as GPSR.

Phase 2. Query execution

Each root node S_i maintains a window, designated W_{2i} , that contains the tuples received from its region. A set of tuples, noted B_{1i} , represents the tuples transmitted periodically by the site S_i . K_{1i} characterizes the projection of B_{1i} on the join attributes.

For a pair of nodes (S_i, S_{i+1}) , an intermediate join is executed as follows (Figure 4):

- i. A projection K_{i+1} is transmitted from S_{i+1} to S_i .
- ii. A semi-join is performed, at S_i , between w_{2i} and K_{i+1} .
- iii. The result $W_{2i'}$ is transmitted to S_{i+1} .
- iv. The final result of the intermediate join is determinate at S_{i+1} .

Phase 3. Final result transmission

With the last intermediate join executed, the determined result is communicated to the sink.

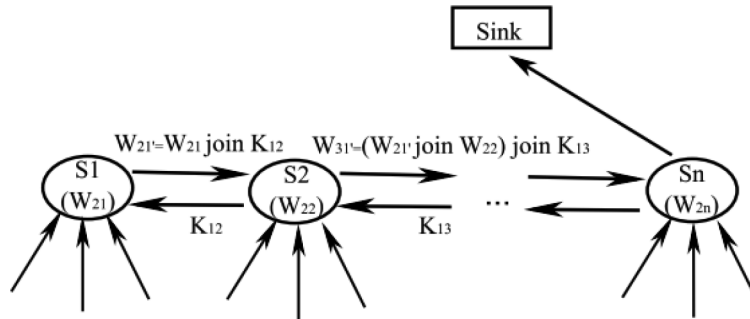


Figure 4. N-way Stream local Semi-Join (NSLSJ) execution

4.4. Sens-join for Data Streams

Sens-join (Stern et al., 2009) is the technique proposed by Stern et al. to treat all types of join queries in wireless sensor networks. Sens-join has five phases:

Phase 1. Query dissemination

A query is initially diffused from the sink to all specified root nodes.

Phase 2: Join attributes transmission

The join attributes are communicated by root nodes to the sink (Figure 5). The aim is to fix the filter of the join query.

Phase 3: Filter determination

The sink produces a filter that is then directed to root nodes in order to perform the semi-join operation.

Phase 4: Semi-join accomplishment

The root nodes perform the semi-join operation. The result is then transmitted to the sink.

Phase 5: Final execution

At the base station, the final result of a join is determined according to the results that have been received from root nodes.

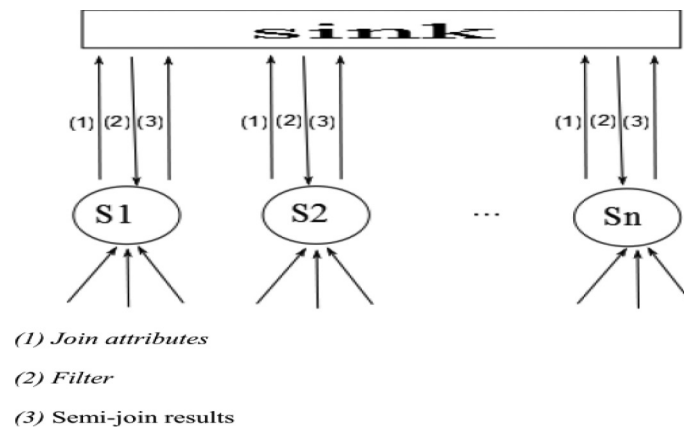


Figure 5. SENS-join execution for n-way stream

5. Experimentation and Performance Analysis

5.1. General Description

A comparative study of the four techniques has been conducted with the aim of determining the technique that performs best and under what conditions of experimentation. This experiment, we consider continuous inter-region joins with the syntax as follows:

```
SELECT S1.attributs, S2.attributs, ..., Sn.attributs  
FROM S1, S2, ..., Sn  
WHERE predicat(S1) AND predicat(S2) ... AND predicat(Sn)  
AND join-exp (S1.join-attributs , S2.join-attributs , ..., Sn.join-attributs)
```

where:

S_i is the stream of the i^{th} region.

predicat (S_i) is a selection predicate of the stream S_i .

join-exp is the join condition.

In this experiment, the example of vehicle traffic control across a variety of geographic areas has been used. For three regions, we write:

```
SELECT Veh1.VId, Veh1.time, Veh2.time, Veh3.time
FROM Veh1, Veh2, Veh3
WHERE (Veh1.time IN r1) and (Veh2.time in r2) and (Veh3.time in r3) and (Veh1.VId = Veh2. VId)
and (Veh2. VId= Veh3. VId)
```

where:

r1, r2, and r3 indicate time ranges during which the Vehicles passed respectively through zones 1,2 and 3.

5.2. Experimentation Environment

In this evaluation, the NS3 simulator is used to simulate an n-way join query. The techniques described above were evaluated by assuming this follows:

- Tuple size is 40 bytes.
- Message size is 40 bytes.
- A Column is 10 bytes.
- The result tuple size is 30 bytes.

As a principal measure, the communication cost was evaluated according to multiple parameters during the actual tests:

- Selectivity factor
- Number of streams
- Size of a stream window.

Selectivity factor values are considered in two intervals: one for low values [10^{-5} , 10^{-4}] and another for high values [10^{-4} , 10^{-3}]. All values are generated randomly. An average value is determined between those engendered for all intermediate joins. A simulation is realized for three streams and then for five streams. The size of stream windows is assumed to equal to 900 tuples.

The number of streams considered in the second evaluation is selected between two and seven. Two evaluations were performed: one with a selectivity factor value in the lowest interval and equal to 0.000025, other with a selectivity factor value in the high interval and equal to 0.00025.

For the third evaluation, stream window sizes are selected between 700 and 1100 tuples. Two simulations were performed with two distinct values of selectivity factor: 0.000025 and 0.00025.

6. Impact of Join Selectivity Factors

In all the performed tests, NSLSJ has performed better than all the other techniques (Figures 6-9). The success of NSLSJ lies in its use of the semi-join principle to realize tuples filtering.

Given specific conditions, the NSLJ technique executes better than SENS-join.

Five streams with low values of selectivity factor up to the value 0.00025, NSLSJ returns low-cost communication.

The performance of Sens-join technique is close to that of NSLSJ. Sens-join is a filtering technique, which is based on the use of filters to remove non-joinable tuples before query execution. The disadvantage of Sens-join is that the query is performed at the sink, but not at the internal nodes of the network.

Extern-join introduces poor results in the tested intervals, but the cost communication remains constant for all the values of all intervals. It has been confirmed that extern-join is attracted in use with very high selectivity factors.

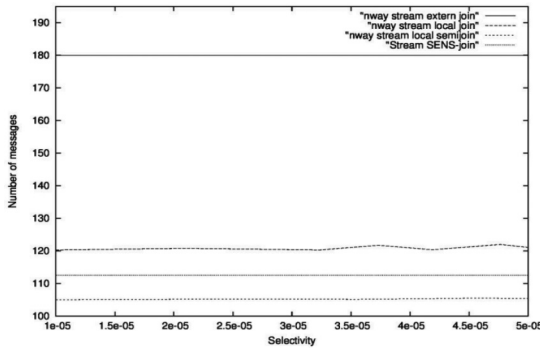


Figure 6. Communication cost for 3 streams in the interval $[10^{-5}, 10^{-4}]$

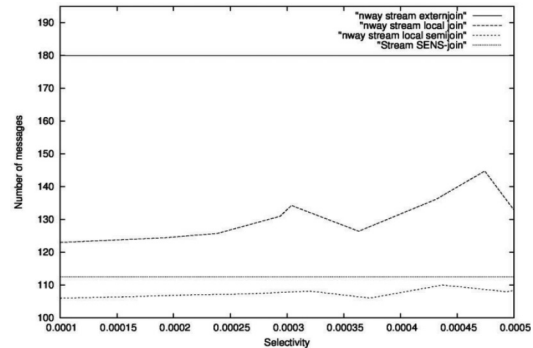


Figure 7. Communication cost for 3 streams in the interval $[10^{-4}, 10^{-3}]$

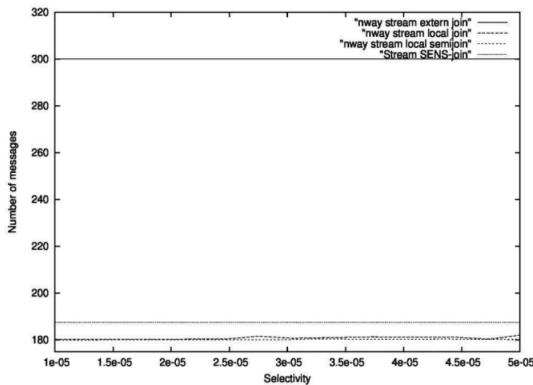


Figure 8. Communication cost for 5 streams in the interval $[10^{-5}, 10^{-4}]$

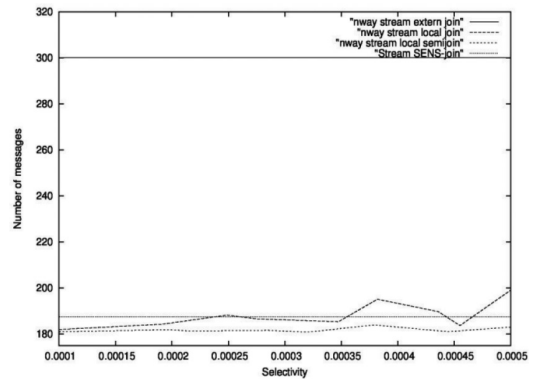


Figure 9. Communication cost for 5 streams in the interval $[10^{-4}, 10^{-3}]$

7. Impact of Number of Streams

NSLSJ is the technique that permits high performances but up to five streams. Beyond this limit, NSLJ is the best (Figures 10-11). This shows that NSLJ, sends fewer messages for a high number of streams.

However, NSLJ provides worse results than NSLSJ and Sens-join for a lower number of streams, specifically, for less than five streams. Sens-join always offers performances close to those of NSLSJ, but it is less efficient. With Extern-join, the number of transmitted messages is very high and grows considerably.

8. Impact of Sizes of Tables

NSLSJ continuously presents the best results, with minor values for the number of the transmitted messages.

SENS-join is less efficient than NSLSJ, but more efficient than NSLJ for all values of the tested intervals (Figures 12-13). Extern-join has poor results; its values are very high and very far from the results of the other tested techniques.

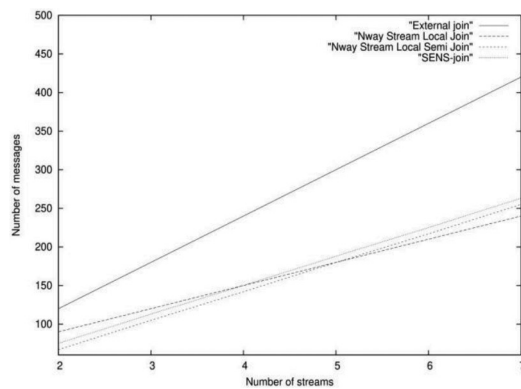


Figure 10. Communication cost depending on streams numbers with selectivity factor equal to 0.000025

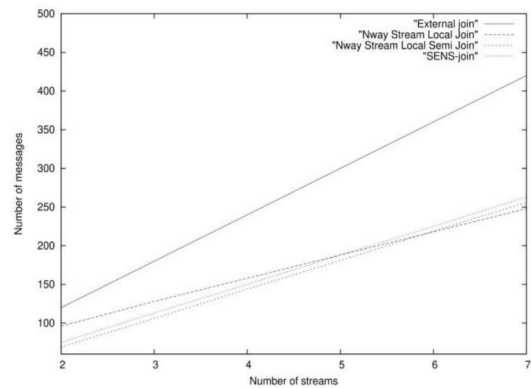


Figure 11. Communication cost depending on streams numbers with selectivity factor equal to 0.00025

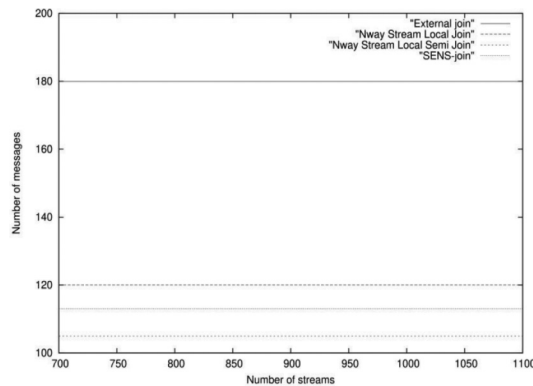


Figure 12. Communication cost depending on windows sizes with selectivity factor equal to 0.000025

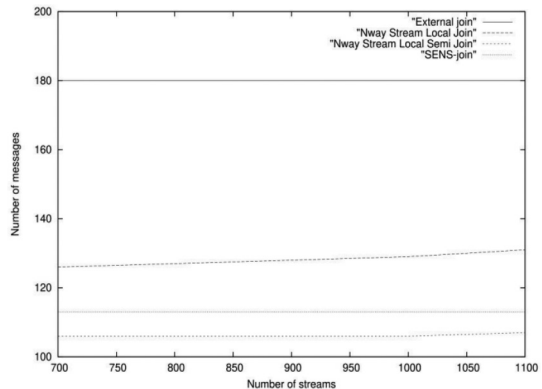


Figure 13. Communication cost depending on windows sizes with selectivity factor equal to 0.00025

8.1. Discussion

All the conducted tests confirm that NSLSJ is the technique that offers the best performance, whether with selectivity factor, number of streams or windows sizes. The efficiency of the NSLSJ lies in its use of the semi-join principle in addition to in-network execution.

Sens-join performances are closer than those of NSLSJ. Sens-join is not as efficient because it uses insufficient filtering to eliminate non-joinable tuples and executes the query totally at the sink.

NSLJ is less than NSLSJ and Sens-join, but for a high number of streams, it offers interesting results. This can be explained by the fact of the reduction of the number of transmitted messages after each execution of an intermediate join.

Extern-join continuously indicates high values of transmitted messages for the selected values of selectivity factor because all concerned tuples must be communicated to the sink where the join query must be performed, which leads to bad performance. Note that Extern-join offers the best execution choice for high value of selectivity factors.

9. Conclusion

Four techniques of n-way stream join queries in the wireless sensor networks have been compared. The first technique is Extern-join, a reference technique in the join execution in wireless sensor networks. The second is Sens-join, an interesting technique with high performance, proposed by Stern et al. The two other techniques have been proposed by Djail et al.

After performing evaluations, NSLSJ showed the best performance due essentially to its adoption of semi-join and the in-networks execution principles. Sens-join has close performances than NSLSJ. However, NSLJ and Extern-join present weak results. A conclusion has been reached that NSLSJ is the best choice for join queries with low join selectivity factors, and extern-join is more accommodating for high values of selectivity factor.

Future research will focus on improving filtering in NSLSJ, to have more performances than those achieved. Join queries for specific join queries in wireless sensor networks will also be studied. Specific join queries in wireless sensor networks will also be addressed. Recent studies in this field are those of Mo et al. (Mo et al., 2014) for spatial queries and Kang (Kang, 2015) for iceberg joins.

References

- Bonfils, B. J., and Bonnet, P. (2004). Adaptive and decentralized operator placement for in-network query processing. *Telecommunication Systems*, 26(2-4), 389-409.
- Boubekeur, D., Khaled, H. W., and Malik, L. (2018). An energy-efficiency technique for n-way stream joins in wireless sensor networks. *Nature & Technology Journal*, <http://www.univ-chlef.dz/revuenatec> Vol. A: Fundamental et Engineering Sciences, 18 (2018), 09-15.
- Coman, A., Nascimento, M. A., and Sander, J. (2007). On join location in sensor networks. Paper presented at the 2007 International Conference on Mobile Data Management.
- Djail, B., Hidouci, K. W., and Loudini, M. (2015). N-way Local Join: a technique for n-way join in wireless sensors networks Paper presented at the International conference on advanced communication systems and signal processing 8-9 November 2015 Tlemcen Algeria.

- Djail, B., Hidouci, K. W., and Loudini, M. (2016). N-way Local SemiJoin : A Filtering Technique for N-Way Joins in Wireless Sensors Networks. *Journal of Electronic Systems*, 6(1), 7-16.
- Djail, B., Hidouci, W.-K., and Loudini, M. (2018). NMSJ: A Filtering Technique for N-Way Joins in Wireless Sensors Networks. *University Politehnica of Bucharest Scientific Bulletin Series c-Electrical Engineering and Computer Science*, 80(4), 23-34.
- Djail, B., Hidouci, W. K., and Loudini, M. (2019). A filtering technique for n-way stream joins in wireless sensors networks. *Revista de Direito, Estado e Telecomunicações*, 11(1).
- Djail, B., Hidouci, W. K., and Loudini, M. (2020). A comparative evaluation of techniques for N-way joins in wireless sensors networks. *Pollack Periodica*, 15(2), 13-24.
- Kang, H. (2013). In-network processing of joins in wireless sensor networks. *Sensors*, 13(3), 3358-3393.
- Kang, H. (2015). In-Network Processing of an Iceberg Join Query in Wireless Sensor Networks Based on 2-Way Fragment Semijoins. *Sensors*, 15(3), 6105-6132.
- Karp, B., and Kung, H.-T. (2000). GPSR: Greedy perimeter stateless routing for wireless networks. Paper presented at the Proceedings of the 6th annual international conference on Mobile computing and networking.
- Min, J.-K., Kim, J., and Shim, K. (2014). TWINS: Efficient time-windowed in-network joins for sensor networks. *Information Sciences*, 263, 87-109.
- Min, J.-K., Yang, H., and Chung, C.-W. (2011). Cost based in-network join strategy in tree routing sensor networks. *Information Sciences*, 181(16), 3443-3458.
- Mo, S., Fan, Y., Li, Y., and Wang, X. (2014). Multi-attribute join query processing in sensor networks. *Journal of Networks*, 9(10), 2702-2712.
- Savvides, A., Srivastava, M., Girod, L., and Estrin, D. (2004). Localization in sensor networks *Wireless sensor networks* (pp. 327-349): Springer.
- Steinbrunn, M., Moerkotte, G., and Kemper, A. (1993). Optimizing join orders: Citeseer.
- Stern, M., Buchmann, E., and Böhm, K. (2009). Towards efficient processing of general-purpose joins in sensor networks. Paper presented at the 2009 IEEE 25th International Conference on Data Engineering.
- Tran, T. M., and Lee, B. S. (2010). Distributed stream join query processing with semijoins. *Distributed and Parallel Databases*, 27(3), 211-254.
- Yao, Y., and Gehrke, J. (2003). Query Processing in Sensor Networks. Paper presented at the CIDR.
- Yu, H., Lim, E.-P., and Zhang, J. (2006). On in-network synopsis join processing for sensor networks. Paper presented at the 7th International Conference on Mobile Data Management (MDM'06).