



# Technical Report

---

## **Efficient Computation of MIN and MAX Sensor Values in Multihop Networks** *by exploiting a prioritized MAC protocol*

Preliminary results of this work were included in "Exploiting a Prioritized MAC Protocol to Efficiently Compute Min and Max in Multihop Networks", presented at the 5th Workshop on Intelligent Solutions in Embedded Systems, June 2007

**Nuno Pereira**  
**Björn Andersson**  
**Eduardo Tovar**  
**Paulo Carvalho**

---

HURRAY-TR-080901

Version: 0

Date: 09-03-2008

# Efficient Computation of MIN and MAX Sensor Values in Multihop Networks *by exploiting a prioritized MAC protocol*

<sup>1</sup>Nuno Pereira, <sup>1</sup>Björn Andersson, <sup>1</sup>Eduardo Tovar, <sup>2</sup>Paulo Carvalho

<sup>1</sup>IPP-HURRAY!

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8340509

E-mail: [nap@isep.ipp.pt](mailto:nap@isep.ipp.pt), [bandersson@dei.isep.ipp.pt](mailto:bandersson@dei.isep.ipp.pt), [emt@isep.ipp.pt](mailto:emt@isep.ipp.pt)

<http://www.hurray.isep.ipp.pt>

<sup>2</sup> Computer Communications Group, Department of Informatics

School of Engineering of the University of Minho

Gualtar, 4710-057 Braga

Portugal

Tel.: +351 253 604 432/6, Fax: +351 253 604 471

E-mail: [pmc@di.uminho.pt](mailto:pmc@di.uminho.pt)

<http://marco.uminho.pt/CCG/>

## Abstract

Consider a wireless sensor network (WSN) where a broadcast from a sensor node does not reach all sensor nodes in the network; such networks are often called multihop networks. Sensor nodes take sensor readings but individual sensor readings are not very important. It is important however to compute aggregated quantities of these sensor readings. The minimum and maximum of all sensor readings at an instant are often interesting because they indicate abnormal behavior, for example if the maximum temperature is very high then it may be that a fire has broken out. We propose an algorithm for computing the min or max of sensor reading in a multihop network. This algorithm has the particularly interesting property of having a time complexity that does not depend on the number of sensor nodes; only the network diameter and the range of the value domain of sensor readings matter.

## Chapter 4

# EFFICIENT COMPUTATION OF MIN AND MAX SENSOR VALUES IN MULTIHOP NETWORKS

*by exploiting a prioritized MAC protocol*

Nuno Pereira<sup>1</sup>, Björn Andersson<sup>1</sup>, Eduardo Tovar<sup>1</sup> and Paulo Carvalho<sup>2</sup>

<sup>1</sup>IPP-HURRAY Research Group, CISTER/ISEP, Polytechnic Institute of Porto, Portugal

<sup>2</sup>Department of Informatics, University of Minho, Braga, Portugal

**Abstract:** Consider a wireless sensor network (WSN) where a broadcast from a sensor node does not reach all sensor nodes in the network; such networks are often called multihop networks. Sensor nodes take sensor readings but individual sensor readings are not very important. It is important however to compute aggregated quantities of these sensor readings. The minimum and maximum of all sensor readings at an instant are often interesting because they indicate abnormal behavior, for example if the maximum temperature is very high then it may be that a fire has broken out. We propose an algorithm for computing the min or max of sensor readings in a multihop network. This algorithm has the particularly interesting property of having a time complexity that does not depend on the number of sensor nodes; only the network diameter and the range of the value domain of sensor readings matter.

**Key words:** Transducers, Data Processing, Large-Scale Sensor Networks, MAC protocol.

## 1. INTRODUCTION

Wireless sensor networks (WSN) often take many sensor readings of the same type (for example, temperature readings), and instead of knowing each individual reading it is important to know aggregated quantities of these sensor readings. For example, each sensor node senses the temperature at its location, and the goal is to know the maximum temperature among all nodes at a given moment.

Several solutions for data aggregation have been proposed for multihop networks. Typically, nodes self-organize into a convergecast tree with a base station at the root [1, 2]. Leaf nodes broadcast their data. All other nodes

wait until they have received a broadcast from all of their children; a node aggregates the data from its children and makes a single broadcast. Techniques have been proposed for computing useful aggregated quantities such as minimum and maximum values, the number of nodes and the median among a set of sensor nodes. They offer good performance because they exploit the opportunities for parallel transmission, and the processing enroute makes the transmitted packet typically smaller than the sum of the size of the incoming packets.

Despite these optimizations, the performance is still inhibited by the fact that in a single broadcast domain, at most one packet can be sent and hence the time-complexity still depends on the number of sensor nodes. This is particularly problematic for dense networks, where even a small broadcast domain (covering an area  $<10m^2$ ) may contain a several tens to a few hundred sensor nodes. In order to improve performance to another level, it is necessary to design distributed algorithms that circumvent this limitation.

In this paper, we propose an algorithm for computing the min or max of sensor readings in a multihop network. This algorithm has the particularly interesting property of having a time complexity that does not depend on the number of sensor nodes; only the network diameter and the range of the value domain of sensor readings matter.

We consider this result to be significant because: (i) a significant number of sensor networks are designed for large scale, dense networks and it is exactly for such scenarios that our algorithms excel and (ii) the techniques that we use depend on the availability of a prioritized MAC protocol that supports a very large range of priority levels and is collision-free assuming that priorities are unique, and such a protocol has recently been proposed, implemented and tested on a sensor network platform [3].

The remainder of this paper is structured as follows. Section 2 starts by providing an introduction on wireless bit dominance and an application background introducing the main idea of how a prioritized MAC protocol can be used, focusing in a single broadcast domain. The final subsection of Section 2 discusses some related work. Section 3 presents the new algorithm which offers a time-complexity that is independent of the number of sensor nodes. Finally, Section 4 draws conclusions.

## 2. PRELIMINARIES AND MOTIVATION

### 2.1 Wireless Bit Dominance

The basic premise for this work is the use of a prioritized MAC protocol for wireless medium. This implies that the MAC protocol assures that of all nodes contending for the medium at a given moment, the ones with the highest priority gain access to it. As a result of the contention for the medium, all participating nodes will have knowledge of the winner's priority. This is inspired by Dominance/Binary-Countdown protocols [4], implemented for wired networks in the widely used CAN bus [5].

In our prioritized MAC protocol for wireless medium, lower priority values mean higher priority, which is also similar to Dominance/Binary-Countdown protocols. However, such protocols assume that priorities are unique. We do not make that assumption.

The protocol in [3] offers this behavior for wireless channels. In this prioritized MAC protocol [3] (inspired by Dominance/Binary-Countdown protocols), the nodes start by agreeing on an instant when the contention resolution phase, named tournament, starts. Then nodes transmit the priority bits starting with the most significant bit. A bit is assigned a time interval. A node contends with a dominant bit ("0"), then a carrier wave is transmitted in this time interval; if the node contends with a recessive bit ("1"), it transmits nothing but listens. At the beginning of the tournament, all nodes have the potential to win, but if a node contends with a recessive bit and perceives a dominant bit then it withdraws from the tournament and cannot win. If a node has lost the tournament then it continues to listen in order to know the priority of the winner. When a node finishes sending all priority bits without hearing a dominant bit when it transmitted a recessive bit, then it has won the tournament and clearly knows the priority of the winner. Hence, lower numbers represent higher priorities. The work developed in [3] includes the precise definition of the protocol timing parameters and accounts for real-world non-idealities such as clock inaccuracies, time of flight, time for detection of carrier pulses or processing delays, and also presents an implementation of the protocol in real-world platforms. While the proof-of-concept implementation of this prioritized MAC protocol for wireless networks introduces a significant amount of overhead, this overhead is, to a large extent, due to the transition time between transmission and reception, which is essentially a technological parameter, as witnessed by the fact that the Hiperlan standard [6] required a switching time of  $2\mu\text{s}$ .

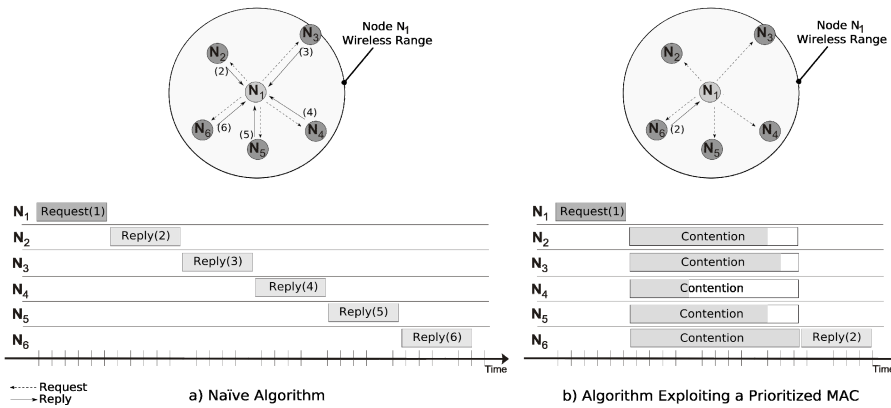


Figure 4-2. Computing min and max in a single broadcast domain.

## 2.2 Motivating Scenario

The focus of this paper will be on exploiting a prioritized MAC protocol as described in the previous subsection. We show that the availability of such a protocol enables efficient distributed computations of aggregated quantities in WSN.

The problem of computing aggregated quantities in a single broadcast domain can be solved with a naïve algorithm: every node broadcasts its sensor reading. Hence all nodes know all sensor readings and then they can compute the aggregated quantity. This has the drawback that in a broadcast domain with  $m$  nodes, at least  $m$  broadcasts are required to be performed. We address the case of WSN designed for large scale, dense networks [7, 8]. Under such premise, the naïve approach can be inefficient, causing a large delay and energy waste.

Let us consider the simple application scenario depicted in Figure 4-1(a), where a node (node  $N_1$ ) needs to know the minimum temperature reading among its neighbors. Let us assume that no other node attempts to access the medium before this node. A naïve approach would imply that  $N_1$  broadcasts a request to all its neighbors and then waits for the corresponding replies from them. As a simplification, assume that nodes have set up a scheme to orderly access the medium in a time division multiple access (TDMA) fashion, and that the initiator node knows the number of neighbor nodes. Then  $N_1$  can compute a waiting timeout for replies based on this knowledge. Clearly, with this approach, the execution time depends on the number of neighbor nodes ( $m$ ).

Consider now that a prioritized MAC protocol such as the one described in the beginning of this section is available. This alternative would allow an approach as depicted in Figure 4-1(b). Assume that the range of the analog to digital converters (ADC) on the sensor nodes is known, and that the MAC protocol can, at least, represent as many priority levels. Now, to compute the minimum temperature among its neighbors, node  $N_1$  needs to perform a broadcast request that will trigger all its neighbors to contend for the medium using the prioritized MAC protocol. If neighbors access the medium using the value of their temperature reading as the priority, the priority winning the contention for the medium will be the minimum temperature reading. (The different lengths of the gray bars inside the boxes depicting the contention in Figure 4-1(b) represent the amount of time that the node actively participated in the medium contention). With this scheme, more than one node can win the contention for the medium. But considering that as a result of the contention, nodes will know the priority of the winner, no more information needs to be transmitted by the winning node.

In this scenario, the time to compute the minimum only depends on the time to perform the contention for the medium, not on  $m$ .

A similar approach can be used to compute the maximum temperature reading. Instead of directly coding the priority with the temperature reading, nodes will use the bitwise negation (change every bit of the temperature reading to its opposite value) of the temperature reading as the priority. Upon completion of the medium access contention, given the winning priority, nodes perform bitwise negation to know the maximum temperature.

### 2.3 Previous work

A prioritized MAC protocol is useful to schedule real-time traffic [3] and it can support data dissemination when topology is unknown [9]. In this paper we have discussed how to efficiently compute aggregated quantities using a prioritized MAC protocol. Distributed calculations have been performed in previous research. It has been observed [10, 11] that nodes often detect an event and then need to spread the knowledge of this event to their neighbors [10]. This is called one-to- $k$  communication [10] because only  $k$  neighbors need to receive the message. After that, the neighbor nodes perform local computations and report back to the node that made the request for 1-to- $k$  communication. This reporting back is called  $k$ -to-1 communication. Algorithms for both 1-to- $k$  and  $k$ -to-1 communication are shown to be faster than a naïve algorithm but, unfortunately, the time-complexity increases as  $k$  increases. On a single broadcast domain, our algorithms compute a function  $f$  and take parameters from different nodes,

making the result available to all nodes. In this respect, it is similar to the average calculations in [12]. However, our algorithms are different from [10-12]; our algorithms have a time-complexity independent of the number of nodes.

One way to use these algorithms is to encapsulate them in a query processor for database queries. Query processors for sensor networks have been studied in previous work [2, 13] but they are different in that they do not compute aggregated quantities as efficiently as we do. They assume one single sink node and that the other nodes should report an aggregated quantity to this sink node. The sink node floods its interest in the data it wants into the network and this also causes nodes to discover the topology. When a node has new data, it broadcasts this data; other nodes hear it, then it is routed and combined so that the sink node receives the aggregated. These works exploit the broadcast characteristics of the wireless medium (like we do) but they do not make any assumption on the MAC protocol (and hence they do not take advantage of the MAC protocol). One important aspect of these protocols is to create a spanning tree. It is known that computing an optimal spanning tree for the case when only a subset of nodes can generate data is equivalent to finding a Steiner-tree, a problem known to be NP-hard (the decision problem is NP-complete, see page 208 in [14]). For this reason, approximation algorithms have been proposed [15, 16]. However, in the average case, very simple randomized algorithms perform well [17]. Since a node will forward its data to the sink using a path which is not necessarily the shortest path to the sink, these protocols cause an extra delay. Hence, there is a trade-off between delay and energy-efficiency. To make this trade-off, a framework based on feedback was developed [18] for computing aggregated quantities. Techniques to aggregate data in the network such that the user at the base station can detect whether one node gives faked data has been addressed as well [19].

Common to these previous works is that the time-complexity increases with the number of sensor nodes.

### 3. THE NEW ALGORITHM

It should be clear that the algorithms for computing min and max in a single broadcast domain (presented in Section 2) do not work in a multihop network. In this section, we will extend it.

We assume that nodes are statically placed in a physical location, and that the communication range ( $R_{co}$ ) is the maximum range at which two nodes  $N_i$  and  $N_j$  can communicate reliably and the interference range ( $R_{it}$ ) is the maximum range between nodes  $N_j$  and  $N_k$  such that simultaneous



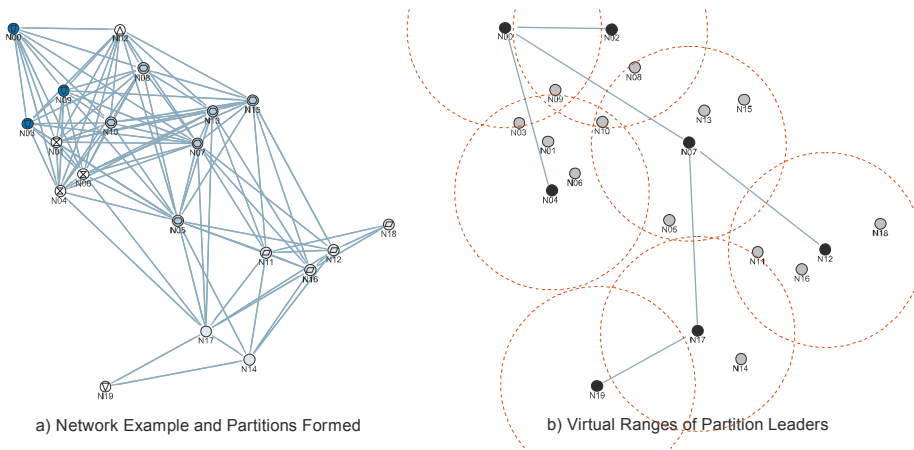


Figure 4-3. Illustration of the MVDS construction algorithm.

transmissions to  $N_j$  will collide with  $N_k$ . We assume that  $R_{it} \leq 2R_{co}$ . We also assume that time is slotted such that all nodes know the time when a timeslot begins and they also know the identifier of the timeslot. One way to implement that is to use a sensor node platform that is equipped with an Amplitude Modulation (AM) receiver that detects signals from an atomic clock. Such AM receivers are used in the FireFly sensor platform [20] and it receives time-sync signals with a continental wide coverage. Two of them are located in Europe; one of them [21] is located in USA. It is assumed that the duration of the timeslot is equal to the time it takes to run a tournament in the MAC protocol. In order to simplify the discussion, we focus on the computation of min of sensor readings; the max of sensor readings can be designed analogously.

It is also assumed that all sensor nodes know when the computation should start. We think the most natural way of doing this is to do it periodically (for example, let all nodes start this computation at the beginning of a timeslot such that the identifier of the timeslot is divisible by 100). This is sensible for applications that continuously detect fire. But in a multi-tiered architecture, where some nodes have a longer communication range, it is possible to let the more high-powered sensor nodes initiate a computation as well; this assumes that those high powered sensor nodes have a communication range that covers the entire network.

The algorithm is composed of two main steps. At setup time, a topology discovery algorithm is executed to partition the network such that all nodes in each partition are in the same broadcast domain. Then, during runtime,

nodes find the minimum sensor reading in all partitions and communicate these values to the leader.

### 3.1 Setup

The setup procedure must partition the network such that (i) each partition forms a single broadcast domain, (ii) a partition leader for each partition is selected, (iii) the partition leaders form a connected distributed set and (iv) to each partition is given a timeslot ensuring that no interfering partitions are active at the same time.

We start this procedure by selecting the partition leaders. To do this we select a *Minimum Virtual Dominating Set* (MVDS) as introduced in [22]. A *Dominating Set* (DS) is a subset of nodes where each node (of the entire graph) is either in the dominating set or is a neighbor to a node in the dominating set. If the set has the minimum cardinality, then it is said to be a *Minimum Set*. To guarantee that all nodes in a partition are in the same broadcast domain, we use a *virtual range*, and thus we construct a MVDS that is the minimum set of nodes required to perform the data aggregation, observing the restrictions (i) to (iii) above.

The details of the algorithm to construct the MVDS can be found in [22]. It is a distributed algorithm with a *propagation phase* that forms the partitions and colors the nodes according to their functionality (*black* if the node is a partition leader or *red* if it is a slave member of a partition), and a *response phase*, where the topology information is delivered to the leader node. In the beginning of the algorithm, all nodes are *white*. The node starting the algorithm (the leader) colors itself black and broadcasts a message with its color. Nodes within the virtual range of the black node become red and nodes that receive the broadcast but are outside the virtual range become blue<sup>1</sup>. After a time interval that is inversely proportional to the distance from the black node, both red and blue nodes forward the message, if they have not done so. Upon being colored, all blue nodes start a timer to become black. This algorithm approximates the solution for a MVDS( $r$ ) composed of the nodes colored black, where  $r$  is the virtual range used. It is important to note that, in this work, we select  $r$  as a function of the communication range such that all nodes in each partition are in the same broadcast domain. Based on our assumptions about the communication range, we can define  $r = R_{co}/2$ .

A possible selection made by the algorithm is illustrated in Figure 4-2. Figure 4-2(a) presents the positions and connectivity of the network. The different partitions formed are also depicted in Figure 4-2(a) by representing

<sup>1</sup> We assume that distances can be approximated; this can be done, for example, using the signal strength in the received packets.



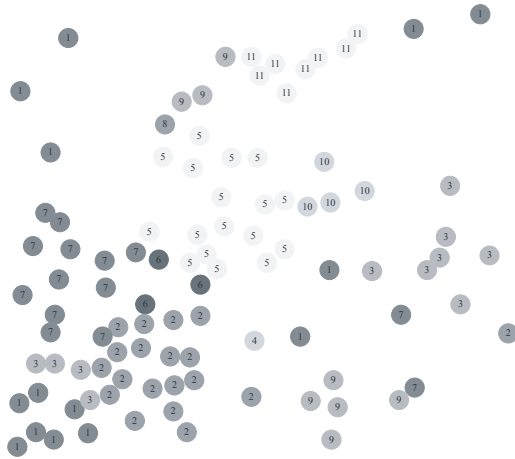


Figure 4-4. Timeslots Assigned to Partitions.

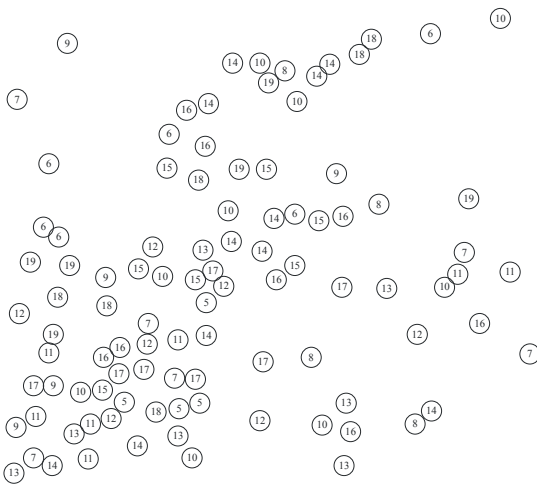


Figure 4-5. Each Sensor Node and the Original Sensor Reading.

Algorithm 1 provides the sequence of steps nodes take during runtime.

While the minimum values are routed to the leader, partition leaders can perform simple processing and avoid forwarding min or max values that are higher or lower than values previously transmitted.

### 3.3 A Running Example

We will illustrate the algorithm with a simple example. Figure 4-3 shows a sensor network consisting of 100 nodes.

Let us consider the algorithm that is run when the sensor network is deployed (as described in Section 3.1). The algorithm partitions the network

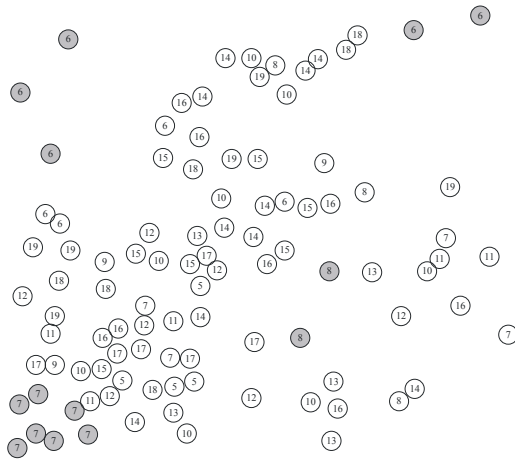


Figure 4-6. Result After Timeslot 1.



Figure 4-7. Result After Timeslot 11.

and selects the corresponding partition leaders. Figure 4-3 depicts the partition leaders with a solid grey circle, the numbers in each node are the partition-ids to which the node belongs (partition-ids are assigned according to the partition leader address).

Then timeslots are assigned to each partition such that if two sensor nodes, in different partitions but in the same timeslot, broadcast simultaneously, then there is no collision. Figure 4-4 shows the timeslot assigned to each node. One can see that there are 11 different timeslots.

Let us consider the algorithm that is executed at runtime. Figure 4-5 shows the temperature readings in all nodes. Nodes compete for the channel using their temperature readings as the priority and nodes do this in their



Figure 4-8. Large-scale Network Example.

assigned timeslot. After this competition, all nodes know the minimum of temperature in the partition. Figure 4-6 shows the result after the first timeslot. Observe that the nodes depicted in solid grey circles have all the same value within the corresponding partitions. This is because these nodes were assigned timeslot 1 and the values depicted are the minimum values in each partition, spread to all sensor nodes in the same partition. After 11 timeslots, all nodes have broadcasted their temperature reading. Figure 4-7 shows the result after the 11:th timeslot. Now, every leader of a partition knows the minimum temperature in the partition. Finally, nodes perform convergecast to the leader of the entire network. Observe that, due to the setup phase, nodes are organized in partitions where member nodes know their partition leaders and partition leaders know the other parent partition leaders who can forward message towards the leader node. Thus performing convergecast is trivial. After the convergecast, the leader knows that the minimum temperature in the entire network is 5.

To further illustrate why the algorithm is fast, a randomly generated network with 1000 nodes is depicted in Figure 4-8. In this figure, the 77 partition leaders are depicted with solid circles, slightly bigger than the other nodes. In this network 17 unique timeslots are needed. By this example, we can observe that our scheme scales well.

So far we have assumed that all transceivers can only transmit in a pre-specified channel. But many wireless standards, such as 802.11, allow a transceiver to transmit on any channel. This feature can be used

advantageously by assigning each partition its own channel (instead of assigning a timeslot to a partition) and this reduces the time required to perform step 2 in Algorithm 1.

#### 4. CONCLUSIONS

We have shown how to use a prioritized MAC protocol to compute aggregated quantities efficiently. The algorithms designed to exploit such MAC protocol have a time-complexity that is independent of the number of sensor nodes. This is clearly important for WSN applications that operate under real-time constraints. But, since the high speed makes it possible for nodes to stay awake for only a short time and they can then sleep, it is also very useful for reducing energy-consumption; and this gives nodes a longer life-time.

We left three important questions open: (i) Can other methods for partitioning the network make this technique perform better? (ii) Can a similar technique be used to compute more complex aggregated quantities (such as COUNT, MEDIAN and interpolation)? (iii) Is the technique sufficiently reliable for large-scale systems?

#### ACKNOWLEDGEMENT

This work was partially funded by CONET, the Cooperating Objects Network of Excellence, funded by the European Commission under FP7 with contract number FP7-2007-2-224053, the ARTIST2 Network of Excellence on Embedded Systems Design and by the Portuguese Science and Technology Foundation (Fundação para Ciência e Tecnologia - FCT).

#### REFERENCES

- [1] J. Gehrke and S. Madden, "Query Processing for Sensor Networks," *IEEE Pervasive Computing*, vol. 3, pp. 46-55, January-March 2004.
- [2] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: a Tiny AGgregation service for ad-hoc sensor networks," in 5th symposium on Operating systems design and implementation (OSDI '02), 2002, pp. 131 - 146
- [3] N. Pereira, B. Andersson, and E. Tovar, "WiDom: A Dominance Protocol for Wireless Medium Access," *IEEE Transactions on Industrial Informatics*, vol. 3, May 2007.
- [4] A. K. Mok and S. Ward, "Distributed Broadcast Channel Access," *Computer Networks*, vol. 3, pp. 327-335, 1979.
- [5] Bosch, "CAN Specification, ver. 2.0, Robert Bosch GmbH, Stuttgart," 1991.

- [6] ETSI, "TS 101 475 V1.3.1.," Broadband Radio Access Networks (BRAN);HIPERLAN Type 2; Physical (PHY) layer.
- [7] A. Arora, "ExScal: Elements of an Extreme Scale Wireless Sensor Network," in Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'05), Washington, DC, USA, 2005, pp. 102-108.
- [8] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," in Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS'00), Maui, U.S.A., 2000, pp. 3005-3014.
- [9] B. Andersson, N. Pereira, and E. Tovar, "Disseminating Data Using Broadcast when Topology is Unknown," in 26th IEEE Real-Time Systems Symposium (RTSS'05), Work-in-Progress Session, 2005, pp. 61-64.
- [10] R. Zheng and L. Sha, "MAC Layer Support for Group Communication in Wireless Sensor Networks," Department of Computer Science, University of Houston UH-CS-05-14, July 21 2005.
- [11] K. Jamieson, H. Balakrishnan, and Y. C. Tay, "Sift: a MAC Protocol for Event-Driven Wireless Sensor Networks," in Third European Workshop on Wireless Sensor Networks (EWSN), Zurich, Switzerland, 2006.
- [12] D. S. Scherber and H. C. Papadopoulos, "Distributed computation of averages over ad hoc networks," IEEE Journal on Selected Areas in Communications, vol. 23, pp. 776- 787, April 2005.
- [13] Y. Yao and J. Gehrke, "Query processing in sensor networks," in Proceedings of the 1st Biennial Conference on Innovative Data Systems Research (CIDR'03), 2003.
- [14] W. Jianping, M. McDonald, M. Brackstone, L. Yangying, and G. Jingjun, "Vehicle to vehicle communication based convoy driving and potential applications of GPS," in Proceedings of the 2nd International Workshop on Autonomous Decentralized Systems, 2002, pp. 212- 217.
- [15] B. Krishnamachari, D. Estrin, and S. B. Wicker, "The Impact of Data Aggregation in Wireless Sensor Networks," in Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02), 2002, pp. 575 - 578.
- [16] C. Intanagonwivat, D. Estrin, R. Govindan, and J. Heidemann, in Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02), Washington, DC, USA, 2002, p. 457.
- [17] E. M., G. A., G. R., and M. R., "Scale-free Aggregation in Sensor Networks," Theoretical Computer Science, vol. 344, pp. 15-29, 2005.
- [18] T. Abdelzaher, T. He, and J. A. Stankovic, "Feedback Control of Data Aggregation in Sensor Networks," in Proceedings of the 43rd IEEE Conference on Decision and Control (CDC'04), 2004, pp. 1490-1495 Vol.2.
- [19] B. Przydatek, D. Song, and A. Perrig, "{SIA}: Secure information aggregation in sensor networks," in Proceedings of the 1st ACM International Conference on Embedded Networked Sensor Systems (SenSys'03), 2003, pp. 255-265.
- [20] R. Mangharam, A. Rowe, and R. Rajkumar, "FireFly: A Cross-Layer Platform for Wireless Sensor Networks," Real Time Systems Journal, Special Issue on Real-Time Wireless Sensor Networks, vol. 37, pp. 183-231, 2007.
- [21] "NIST Radio Station WWVB," <http://tf.nist.gov/stations/wwvb.htm>.
- [22] B. Deb, S. Bhatnagar, and B. Nath, "Multi-resolution state retrieval in sensor networks," in Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on, 2003, pp. 19-29.