CISTER
Research Center in
Real-Time & Embedded
Computing Systems

Technical Report

How many are you (an approach for the smart dust world)?

Michele Albano
Nuno Pereira
Eduardo Tovar

CISTER-TR-130802

2013/08/19

How many are you (an approach for the smart dust world)?

Michele Albano, Nuno Pereira, Eduardo Tovar

CISTER Research Center

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: mialb@isep.ipp.pt, nap@isep.ipp.pt, emt@isep.ipp.pt

http://www.cister.isep.ipp.pt

Abstract

As the size and cost of embedded devices continue to decrease, it becomes economically feasible to densely deploy networks with very large quantities of such nodes, and thus enabling the implementation of networks with increasingly larger number of nodes becomes a relevant problem. In this paper we describe a novel algorithm to obtain the number of live nodes with a very low time-complexity. In particular, we develop a mechanism to estimate the number of nodes or the number of proposed values (COUNT), with a time complexity that increases sublinearly with the number of nodes. The approach we propose is based on the wise exploitation of dominance-based protocols and offers excellent scalability properties for emerging applications in dense Cyber Physical Systems.

How many are you
(an approach for the smart dust world)?

Michele Albano, Nuno Pereira and Eduardo Tovar
CISTER/INESC-TEC, ISEP,
Polytechnic Institute of Porto, Portugal
[mialb,nap,emt]@isep.ipp.pt

***Abstract*—As the size and cost of embedded devices continue to decrease, it becomes economically feasible to densely deploy networks with very large quantities of such nodes, and thus enabling the implementation of networks with increasingly larger number of nodes becomes a relevant problem. In this paper we describe a novel algorithm to obtain the number of live nodes with a very low time-complexity. In particular, we develop a mechanism to estimate the number of nodes or the number of proposed values (COUNT), with a time complexity that increases sublinearly with the number of nodes. The approach we propose is based on the wise exploitation of dominance-based protocols and offers excellent scalability properties for emerging applications in dense Cyber Physical Systems.**

***Index Terms*—Distributed Cooperative Computing; MAC Protocols; Dominance**

I. INTRODUCTION

Nowadays, computing systems and in particular embedded systems are part of our day-to-day life. The number of computing elements we interact with on a daily basis is growing, since domotics technologies got to sufficient maturity to be employed in many residential environments. Additionally, networks with more than one thousand embedded nodes [3] have been deployed for collaborative processing of physical information, and it can be expected that networks with hundreds of thousands of nodes will be deployed within a few years. Large-scale, sensor-rich networked systems will generate an enormous amount of data. The data processing challenges arising from the existence of large networks can be exemplified by considering that computing simple aggregates, such as MIN, MAX or COUNT (the number of nodes that are characterized by a given condition), might require communicating with a large number of nodes, and thus, such computations do not scale well when the number of nodes gets large.

One approach to overcome the problem is to parallelize the communications between the nodes, by leveraging on the spatial decoupling of the communication medium. For example, for multihop networks, several techniques for computing useful aggregated quantities that offer good performance have been proposed [19], [11]. Nonetheless, new challenges arise from scenarios where the communication medium is shared and contains several tens of nodes. In the extreme case where all nodes are in the same broadcast domain, nodes cannot transmit in parallel, and there are no opportunities to parallelize communications and apply traditional data aggregation techniques.

A disrupting approach is given by designing algorithms that take advantage of dominance-based MAC protocols. Because such MAC protocols resolve contentions in a non-destructive manner, they allow more than one node to occupy the medium at the same time and thus can be exploited to compute aggregate values between the nodes participating in the MAC protocol contention. This basic mechanism, inspired by CAN [4], was proposed for Wireless Sensor Networks in [2], and was used in a deterministic algorithm to compute the MIN function over a number of sensor readings [15], [16].

This work performs an abstraction step by considering general Dominance-based communication, and it extends the computation of aggregates with the use of probabilistic algorithms. In particular, we develop a mechanism to estimate the number of nodes or the number of proposed values (COUNT), with a time complexity that is independent from the actual number of nodes. The estimation of COUNT is a relevant building block for other algorithms; for example it can be used to implement voting. One useful example is an application that wants to know the number of nodes that have a sensor reading inside a given range. The approach assumes that all nodes are in the same broadcast domain. However, local aggregation between nodes in geographic proximity can be used as an intermediate step for obtaining aggregated values in multihop networks and/or in hierarchical topologies.

The rest of this paper is organized as follows: Section II provides background material and some recent results on the use of dominance-based protocols for computation of aggregates; Section III describes the assumptions of our work; Section IV presents the COUNT algorithm, and it delves into its analysis, in particular by evaluating its statistical error against the number of iterations performed while executing the algorithm; Section V provides simulation results to corroborate our findings; finally, Section VI wraps up the discussion and draws conclusions.

II. RELATED WORK

The research work described in this paper considers designing algorithms that exploit the dominance paradigm for efficient data aggregation. Before presenting the focus of this paper (randomized algorithms to compute aggregates), we will briefly look into dominance-based MAC protocols and some related results (Subsection II-A). Later, in Subsection II-B, we will review previous research on estimating COUNT.

A. *Exploiting Dominance-Based MAC Protocols*

Let us consider the example of computing MIN in a network with N nodes, where each node has a k-bit temperature sensor. Computing MIN in a traditional way would imply that all N individual sensor readings are compared. Generally, it will take $O(N)$ message transmissions and, due to packet collisions, we cannot assume to transmit all N messages simultaneously.

A dominance-based MAC protocol can be used to efficiently obtain MIN as follows. Starting with the most significant bit first, let each node send the temperature reading bit-by-bit. The transmission of bits is arranged such that, at the end of the transmission of k bits, the "observed" value in the channel will correspond to the MIN of all transmitted values. The time complexity is independent from the number of nodes in the broadcast domain N, and it is linear in the number of bits (k) used to encode the priority value.

To implement such approach, various obstacles (such as how the actual transmission of bits occurs and how nodes can correctly perceive the bit in the same manner), need to be sorted out. In the wired domain, dominance protocols [12] have been implemented in the widely used CAN bus [4]. In [1], the authors demonstrated that CAN-enabled platforms can be used to compute various aggregate quantities (MIN, MAX and Interpolation). The WiDom protocol [2] extends dominance protocols to wireless networks, thus enabling implementing the same behaviour in wireless networks. In this paper, we extend this work by introducing probabilistic algorithms to estimate COUNT.

B. *Previous Work on COUNT*

The problem of obtaining the number of nodes (COUNT) in a network can be viewed from different perspectives. Gossip, rumour spreading and infectious algorithms, all have in common that they use randomized local computations repeatedly to achieve a global computation. Originally these algorithms were developed to propagate data, but later they also have been reworked to calculate aggregated quantities. These algorithms are robust in face of node and link failures and they can operate in multihop networks. Such algorithms are available for a large number of distributed estimates/calculations, such as COUNT and MIN (see for example [8], [7]). The statistical technique used in [5] to estimate COUNT in a peer-to-peer network is similar to our one, but we develop it further in the direction of making it faster using a prior estimation of the number of peers in a network.

Deterministic algorithms for unstructured environments have also been proposed. The algorithm in [18] performs repeated local operations to compute an average, and it works efficiently in multihop environments since it parallelizes the communications over the area. The algorithm in [9] computes the average in a single-hop network. It is designed to perform well against an adversary that injects faults but, unfortunately, its time complexity is very high. These techniques that compute average values could be used to compute the number of nodes. Data aggregation protocols for Wireless Sensor Networks can compute COUNT, typically using a convergecast tree [19], [10]. The same problem has been addressed by researchers in data communications with the goal of estimating the size of the audience of a multicast communication [6], [13]. Anyway, all previous algorithms assume the presence of spatial decoupling, hence the parallelization of the communication. Common to all these works is that their time complexity grows linearly with the number of nodes communicating in the shared medium N (i.e. O(N)) or more, whereas our technique has a time complexity which increases sublinearly with N.

Algorithm 1 Estimating COUNT (the number of nodes in the network)

```
Require: All computer nodes start their execution simultaneously.
Require: active is a boolean variable specifying if the node is participating in the tournament.
r : array[1..R] of integer
x : array[1..R] of integer
q : integer
for $q \leftarrow 1$ to $R$
    $r[q] \leftarrow random(1, M)$
    if ($active$==TRUE) then $x[q] \leftarrow send\_empty(r[q])$
    else $x[q] \leftarrow M$
end for
$est\_nodes \leftarrow ML\_estimation(M; [x[1], x[2], ..., x[R]])$
return $est\_nodes$ // the estimation of the number of nodes
```

III. SYSTEM MODEL

Consider a system comprising N nodes that can communicate by broadcast, both in the traditional way (one node sending a message) and with a dominance-based MAC protocol as described earlier. It is assumed that every transmitted signal is received by all nodes, and this implies that there are no hidden stations and the network provides reliable broadcast. We also assume that a protocol is in place to ensure that, during the estimation of COUNT, nodes do not transmit any other kind of messages. It is also assumed that nodes are commanded to compute the number of nodes simultaneously. Afterwards, the MAC protocol will take care of keeping the nodes in sync, with techniques that are dependent on the involved technology (see for example CAN[4] and WiDom[2]).

Let k denote the number of priority bits that all nodes employ while trying to get access to the medium. The function that generates the random number is denoted by `random`$(1, M)$, and it outputs random numbers in the range $[1, M = 2^k]$.

The communication stack offers the following calls for interacting with other nodes. Function `send` takes two parameters, one describing the priority of the message and one describing the data bits to be transmitted. If `send`'s priority doesn't grant priority to a node, the program invoking the system call blocks until the message is successfully transmitted, possibly after trying to access the medium a number of times. The function `send_empty` takes only one parameter, which is the priority of the node calling the function. A call

Algorithm 2 Function `ML_estimation`

Require: The division of two integers (as is done in line 6) returns a real number.

```
1: function ML_estimation($M$: integer; $x$ : array[1..$R$] of integer) return an integer
2:     $v$ : array[1..$R$] of real
3:     $sumv$, $q$ : integer
4:     $sumv \leftarrow 0$
5:     for $q \leftarrow 1$ to $R$
6:         $sumv \leftarrow sumv + \frac{v[q]}{R}$
7:     end for
8:     return $floor(\ M\ /\ sumv\ ) - 1$
9: end function
```

to `send_empty` results in the MAC protocol performing the contention for the medium, but if the node wins, it does not send anything. In addition, after the execution of the protocol, `send_empty` gives the control back to the application and returns the priority of the winner, regardless of whether the node won or lost the contention. The `send_empty` can be used in environments where two nodes may have the same priority and hence there may be more than one node that declares itself as a winner. This is acceptable since they do not send any data, so the mechanism causes no data collision.

IV. ESTIMATING COUNT BY USING A RANDOMIZED ALGORITHM

This section introduces a randomized algorithm to estimate the number of the nodes in the area. In particular, Subsection IV-A introduces the basic COUNT algorithm, which is the basis for the analysis and the enhancements presented in the rest of the section. The intuition behind our method is as follows: if the value used for the dominance-based MAC protocol contention by each node is a positive number chosen uniformly at random in $[1, M]$, then the probability for the minimum value of the contention fields to be small approaches 1 as the number of nodes gets larger.

Later on in this section, Subsection IV-B provides a rationale for the design of the functions underlying the COUNT algorithm by performing an analysis of the minimum random number that is expected to be collected.

Subsection IV-C extends the analysis by providing an analytical formulation of the error in the COUNT estimation, expressed as a function of the number of rounds R that the dominance protocol is repeated; moreover, the subsection uses the estimated error to suggest the number of repetitions of the dominance protocol to attain a given precision, and estimates the completion time of the COUNT estimation.

Please take into account that some of the analytical results consider a real random number in the range $[0, M]$, while the algorithm uses an integer in $[1, M]$. This is the result of the discretization of the technique, which imposes to disregard either 0 or M. We chose to disregard 0, since the estimation formula (Eq.1) inverts the mean random number hence it is more stable when we disregard 0, and simulations compared the results of disregarding 0 and M and validated our choice.

A. Basic COUNT Algorithm

The pseudo code of the algorithm for estimating the number of nodes is shown in Algorithms 1 and 2. Locally, each node has a boolean variable $active$, which indicates if the nodes should be included in the counting or not. This can be used, for example, if it is desired to count only the nodes with a certain local attribute (e.g. only the nodes measuring a temperature above a certain threshold).

The main algorithm (Algorithm 1) assumes that all nodes start their execution simultaneously and, on line 5, nodes generate a random number in the range $[1, M]$. Thereafter, all nodes contend for the medium using their random number as their priorities, and the `send_empty` function reports the minimum random number (line 6). This is performed for R rounds, to fight the error introduced by the stochastic process using statistic significance. Line 8 computes the estimation of the number of nodes based on the minimum obtained on line 6, by using the function shown in Algorithm 2. The design of the function in Algorithm 2 is based on Bayesian statistics and the rationale behind its design are analyzed in Subsection IV-B and Subsection IV-C.

B. Probability Distribution for the Minimum Random Priority

The parameters that concur to each single dominance round are the number of nodes N and the maximum random number M that can be generated. Apart from that, the precision of the COUNT algorithm depends on the number of rounds R that we use for the estimation of the number of nodes. The analysis presented in this subsection is applied to a single dominance round q. Let us call r_i^q the random number provided by node i during the round q of the algorithm, and let x_q be the minimum number collected at round q.

Given that in round q every node i chooses a number r_i uniformly at random, the probability that the minimum number for that round is x_q can be written as:

$$
\begin{aligned}
&P(\min\{r_1^q ... r_N^q\} = x_q) = \\
&P(r_1^q = x_q \wedge r_1^q \leq r_2^q \wedge ... \wedge r_1^q \leq r_N^q) + \\
&P(r_2^q = x_q \wedge r_2^q \leq r_1^q \wedge ... \wedge r_2^q \leq r_N^q) + ... + \\
&P(r_N^q = x_q \wedge r_N^q \leq r_1^q \wedge ... \wedge r_N^q \leq r_{N-1}^q)
\end{aligned}
$$

For symmetry reasons:

$$
\begin{aligned}
&P(\min\{r_1^q ... r_N^q\} = x_q) = \\
&N \cdot P(r_1^q = x_q \wedge r_1^q \leq r_2^q \wedge ... \wedge r_1^q \leq r_N^q) = \\
&N \cdot P(r_1^q = x_q) \cdot P(r_1^q \leq r_2^q \wedge ... \wedge r_1^q \leq r_N^q)
\end{aligned}
$$

Given that the random numbers are chosen uniformly at random, we have that the probability that x_q is larger than r_1^q is:

$$
P(r_1^q \geq x_q) = \frac{M - x_q}{M} = 1 - \frac{x_q}{M}
$$

and this leads to:

$$P(\min\{r_1^q...r_N^q\} = x_q) =$$
$$\frac{N}{M}\int_0^M \delta(r_1^q - x_q)\left(\frac{M - r_1^q}{M}\right)^{N-1} dx_q =$$
$$\frac{N}{M}\left(1 - \frac{x_q}{M}\right)^{N-1}$$

C. An Estimation for N

Basic algebraic manipulations show that $x_q/M = (1 - (1 - x_q/M))$, which leads to an expected value for $x_q = \min\{r_1^q...r_N^q\}$ that is

$$\mathbf{E}[x_q] = \int_0^M x_q P(\min\{r_1^q...r_N^q\} = x_q)dx_q =$$
$$\int_0^M x_q \tfrac{N}{M}(1 - \tfrac{x_q}{M})^{N-1}dx_q =$$
$$-(N\tfrac{-M}{N} + N\tfrac{M}{N+1}) = M(1 - \tfrac{N}{N+1}) = \tfrac{M}{N+1}$$

Since the “central limit theorem” states that the mean value of a collection of $\{x_1, ..., x_R\}$ approaches $\mathbf{E}[x_q]$, a precise calculation of the mean value for x_q, possible by repeating the CAN algorithm a number of times, can help to invert the formula and compute $N = \frac{M}{\mathbf{E}[x_q]} - 1$ and compute j'', which is an estimation of N:

$$j'' = \frac{M}{\bar{x}} - 1 \qquad (1)$$

where $\bar{x}$ is the mean value for x_q computed over a number of rounds R. Since $\frac{(x_q)^2}{M^2} = (1 - 2(1 - \frac{x_q}{M}) + (1 - \frac{x_q}{M})^2$, the expected value for $(x_q)^2$ is:

$$\mathbf{E}[(x_q)^2] = \int_0^M (x_q)^2 P(\min\{r_1^q...r_N^q\} = x_q)dx_q =$$
$$MN\int_0^M \left[(1 - \tfrac{x_q}{M})^{N-1} - 2(1 - \tfrac{x_q}{M})^N + (1 - \tfrac{x_q}{M})^{N+1}\right]dx_q =$$
$$\frac{2M^2}{(N+1)(N+2)}$$

Thus,

$$\sigma^2(x_q) = \mathbf{E}[x_q^2] - \mathbf{E}[x_q]^2 = \frac{M^2 N}{(N+1)^2(N+2)}$$

and, for a large number of rounds R, the standard deviation of the minimum random numbers r_i^q approaches:

$$\sigma(x_q) = \sqrt{\frac{M^2 N}{(N+1)^2(N+2)}} \approx \frac{M}{N}$$

Given that we execute a sufficient number R of dominance rounds, the mean difference between x_q and $\bar{x}$ computed over R rounds approaches its expected value $\sigma(x_q)$, and the most probable error on j'' (Eq. 1) can be estimated as:

$$\Delta j'' = \left|\frac{M}{\mathbf{E}[x] + (\bar{x_q} - \mathbf{E}[x])} - \frac{M}{\mathbf{E}[x]}\right| \approx \left|\frac{-M(x_q - \mathbf{E}[x])}{(\mathbf{E}[x])^2}\right|$$

The expected difference between $\mathbf{E}[x]$ and $\bar{x}$ computed over a number of rounds R approaches $\frac{\sigma(x_q)}{\sqrt{R}}$ and for $R \gg 1$ the expression converges to:

$$\Delta j'' \approx \left|\frac{M\sigma(x_q)/\sqrt{R}}{(M/N)^2}\right| \approx \frac{N}{\sqrt{R}} \qquad (2)$$

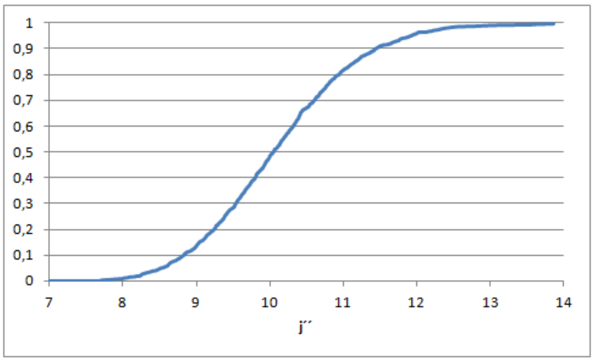

Fig. 1. Cumulative distribution function for the estimation, 10 nodes, 100 rounds per experiment.

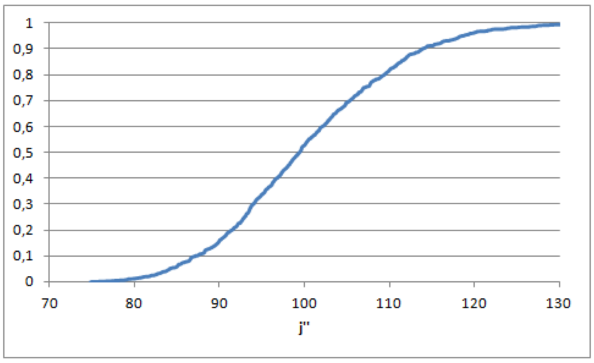

Fig. 2. Cumulative distribution function for the estimation, 100 nodes, 100 rounds per experiment.

This result can be used to estimate the number of rounds R we should repeat the dominance protocol, with the goal of having an expected error of a given magnitude.

V. PERFORMANCE EVALUATION

The evaluation of this section addresses the performance of the proposed approach, and validates the analytical results of Section IV via simulations.

A montecarlo simulator was implemented to verify (i) the correct estimation of the number of nodes, and (ii) prediction of the error on the estimation. We have simulated three scenarios, characterized by different number of nodes ($N = 10$ and $N = 100$), and with the dominance algorithm repeated for a different number of rounds ($R = 100$ and $R = 10000$). For each scenario, we performed 1000 simulations, and every time the pool of random number was $[1, 10^6]$ (so that $\frac{x}{M} << 1$)

The numerical results for the mean values and standard deviation were respectively 10.10 ± 1.02, 101.1 ± 10.3, and 100.42 ± 0.94. Thus, the standard deviations computed by the montecarlo simulations validate that the expected divergence from the real number of nodes N agrees with Eq. 2.

The figures 1, 2 and 3 provide a graphical representation of the results. In each figure we reported the cumulative distribution function for j''. By observing the plots, it is possible to observe that the estimation of the number of nodes

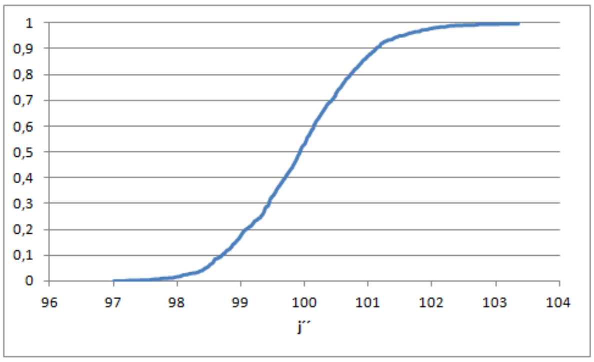

Fig. 3. Cumulative distribution function for the estimation, 100 nodes, 10000 rounds per experiment.

performs quite well. For example, with $N = 10$ and $R = 100$ (Figure 1), almost 70% of the estimates fall between 9 and 11. With $N = 100$ and $R = 100$ (Figure 2), we have about 55% of the estimates in the range $[90, 110[$.

VI. CONCLUSION

This paper proposed a technique for efficiently estimating COUNT in dense networks, whose time-complexity does not depend on the number of nodes. The results of this paper are based on a thorough analysis of the error introduced by the randomized algorithm the mechanism is based on. The contribution to the state-of-the-art is an efficient algorithm to speed-up the computation of the COUNT, and simulation of the execution time shows that the technique speeds up the COUNT estimation.

The technique is significant in a setting where (i) networks are characterized by large scale, high density, and (ii) a prioritized MAC protocol based on the dominance paradigm is available. The technique relies on its priority MAC protocol to support a very large range of priority levels and to be collision-free, and such a protocol has recently been proposed, implemented and tested [2], [15]. Moreover, our technique requires that no faults occur; in general this is difficult to achieve in real networks, and especially when communicating over a wireless medium. However, it was observed that in short distance communication, using a spread spectrum transceiver, it is possible to achieve good reliability [14], [15]. Moreover, techniques to improve the reliability of WiDom have been recently proposed [17].

Future works include the implementation of the algorithm on real nodes employing WiDom, and the extension of the results to a scenario comprising very dense multi-hop networks.

ACKNOWLEDGMENT

This work was partially supported by National Funds through FCT (Portuguese Foundation for Science and Technology) and by ERDF (European Regional Development Fund) through COMPETE (Operational Programme 'Thematic Factors of Competitiveness'), within project Ref. FCOMP-01-0124-FEDER-022701 (CISTER), FCOMP-01-0124-FEDER-020312 (SMARTSKIN) and FCOMP-01-0124-FEDER-012988 (SENODS); also by the project ref. NORTE-07-0124-FEDER-000063 (BEST-CASE), co-funded by ERDF (European Regional Development Fund), through ON2 - North Portugal Regional Operational Programme, under the National Strategic Reference Framework (NSRF).

REFERENCES

[1] B. Andersson, N. Pereira, W. Elmenreich, E. Tovar, F. Pacheco, and N. Cruz. A scalable and efficient approach to obtain measurements in CAN-based control systems. In *IEEE Transactions on Industrial Informatics*, May, 2008.

[2] B. Andersson, N. Pereira, and E. Tovar. Widom: A dominance protocol for wireless medium access. *IEEE Transactions on Industrial Informatics, vol. 3(2)*, May 2007.

[3] 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)*, pages 102–108, Washington, DC, USA, 2005. IEEE Computer Society.

[4] Bosch GmbH, Stuttgart, Germany. *CAN Specification, ver. 2.0*, 1991.

[5] D. V. H. Terelius and K. H. Johansson. Distributed size estimation of dynamic anonymous networks. In *Proc. of IEEE CDC 2012*, Maui, HI, USA, 2012.

[6] K. Horowitz and D. Malkhi. Estimating network size from local information. *Information Processing Letters*, 88(5):237–243, 2003.

[7] M. Jelasity, A. Montresor, and O. Babaoglu. Gossip-based aggregation in large dynamic networks. *ACM Transactions on Computer Systems*, 23(3):219–252, 2005.

[8] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS'03)*, pages 482–491, Washington, DC, USA, 2003. IEEE Computer Society.

[9] M. Kutylwski and D. Letkiewicz. Computing average value in ad hoc networks. In *Proceedings of the 28th International Symposium on Mathematical Foundations of Computer Science (MFCS'03)*, pages 511–520. Springer, 2003.

[10] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. TAG: a tiny aggregation service for ad-hoc sensor networks. In *Proceedings of the 5th symposium on Operating systems design and implementation (OSDI'02)*, 2002.

[11] S. Madden, M. J. Franklin, J. Hellerstein, and W. Hong. TAG: a tiny aggregation service for ad-hoc sensor networks. In *Proceedings of the 5th symposium on Operating systems design and implementation (OSDI'02)*, 2002.

[12] A. K. Mok and S. Ward. Distributed broadcast channel access. *Computer Networks*, 3:327–335, 1979.

[13] M. Nekovee, A. Soppera, and T. Burbridge. An adaptive method for dynamic audience size estimation in multicast. In *Lecture Notes in Computer Science*, volume 2816, pages 23–33, 2003.

[14] N. Pereira, B. Andersson, and E. Tovar. Implementation of a dominance protocol for wireless medium access. In *Proceedings of the 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'06)*, Sydney, Australia, 2006.

[15] N. Pereira, B. Andersson, and E. Tovar. Efficient aggregate computations in large-scale dense WSN. In *15th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'09)*, pages 317–326, San Francisco, California, USA, 2009.

[16] N. Pereira, B. Andersson, E. Tovar, and P. Carvalho. Efficient computation of min and max sensor values in multihop networks - by exploiting a prioritized MAC protocol. *Springer Lecture Notes on Electrical Engineering (LNEE)*, 38, Intelligent Technical Systems:233–246, 2009.

[17] N. Pereira, B. Andersson, E. Tovar, and P. Carvalho. Improving the reliability of widom in a single broadcast domain. In *IEEE Symposium on Industrial Embedded Systems (SIES'09), Work-in-Progress Session*, Lausanne, Switzerland, 2009.

[18] D. Scherber and H. Papadopoulos. Distributed computation of averages over ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(4):776– 787, 2005.

[19] 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.