Technical Report

Efficient Aggregate Computations in Large-Scale Dense WSN

Nuno Pereira, Ricardo Gomes,
Björn Andersson, Eduardo Tovar

HURRAY-TR-081101
Version: 1
Date: 02-13-2009

Efficient Aggregate Computations in Large-Scale Dense WSN

Nuno Pereira, Ricardo Gomes, Björn Andersson, Eduardo Tovar

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:

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

Abstract

We focus on large-scale and dense deeply embedded systemswhere, due to the large amount of information generatedby all nodes, even simple aggregate computations suchas the minimum value (MIN) of the sensor readings becomenotoriously expensive to obtain. Recent research has exploiteda dominance-based medium access control (MAC)protocol, the CAN bus, for computing aggregated quantitiesin wired systems. For example, MIN can be computedefficiently and an interpolation function which approximatessensor data in an area can be obtained efficiently aswell. Dominance-based MAC protocols have recently beenproposed for wireless channels and these protocols can beexpected to be used for achieving highly scalable aggregatecomputations in wireless systems. But no experimentaldemonstration of that is currently available in the researchliterature.In this paper, we demonstrate that highly scalable aggregatecomputations in wireless networks are possible. We doso by (i) building a new wireless hardware platform with appropriatecharacteristics for making dominance-based MACprotocols efficient, (ii) implementing dominance-based MACprotocols on this platform, (iii) implementing distributed algorithmsfor aggregate computations (MIN,MAX, Interpolation)using the new implementation of the dominance-basedMAC protocol and (iv) performing experiments to prove thatsuch highly scalable aggregate computations in wireless networksare possible.

Efficient Aggregate Computations in Large-Scale Dense WSN

Nuno Pereira, Ricardo Gomes, Björn Andersson, Eduardo Tovar
IPP-HURRAY Research Group
CISTER/ISEP, Polytechnic Institute of Porto, Porto, Portugal
nap@isep.ipp.pt, rftg@isep.ipp.pt, bandersson@dei.isep.ipp.pt, emt@dei.isep.ipp.pt

Abstract

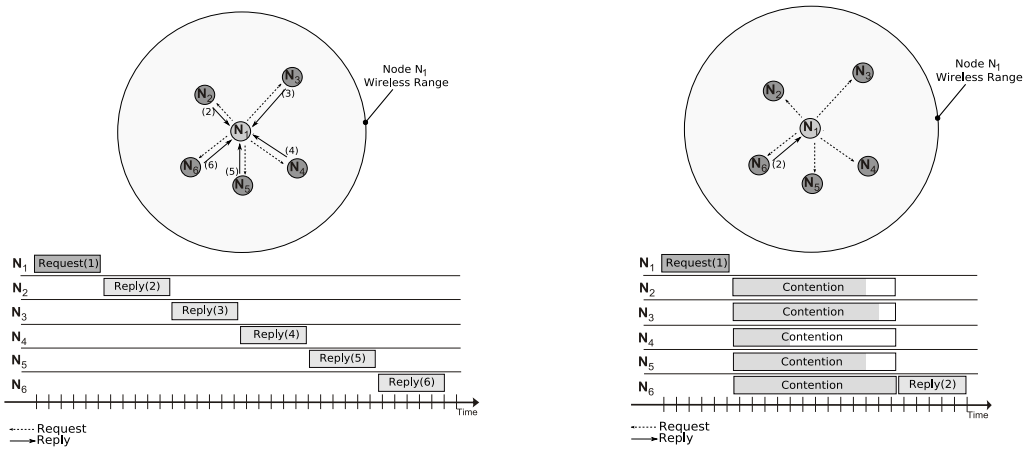
We focus on large-scale and dense deeply embedded systems where, due to the large amount of information generated by all nodes, even simple aggregate computations such as the minimum value (MIN) of the sensor readings become notoriously expensive to obtain. Recent research has exploited a dominance-based medium access control (MAC) protocol, the CAN bus, for computing aggregated quantities in wired systems. For example, MIN can be computed efficiently and an interpolation function which approximates sensor data in an area can be obtained efficiently as well. Dominance-based MAC protocols have recently been proposed for wireless channels and these protocols can be expected to be used for achieving highly scalable aggregate computations in wireless systems. But no experimental demonstration is currently available in the research literature.

In this paper, we demonstrate that highly scalable aggregate computations in wireless networks are possible. We do so by (i) building a new wireless hardware platform with appropriate characteristics for making dominance-based MAC protocols efficient, (ii) implementing dominance-based MAC protocols on this platform, (iii) implementing distributed algorithms for aggregate computations (MIN, MAX, Interpolation) using the new implementation of the dominance-based MAC protocol and (iv) performing experiments to prove that such highly scalable aggregate computations in wireless networks are possible.

1 INTRODUCTION

Interlinking the real-world and the cyberspace efficiently has been the subject of significant research interest [1]. Integration of physical processes and computing is not new; embedded systems have been in place long ago and these often combine physical processes with computing. A distinguishing feature of these new systems however comes from massively networked embedded computing devices, which will allow instrumenting the physical world with pervasive networks of sensor-rich embedded computation [2].

Recently, networks with more than a thousand sensor nodes [3] have been deployed for collaborative processing of physical information, and one may expect that the networks to be deployed within the



(a) Naïve Algorithm

(b) Algorithm Exploiting a Dominance MAC

Fig. 1. Motivating Examples.

next decades will be much denser in order to obtain a more accurate estimate of physical processes. Such large-scale, sensor-rich networked systems will generate an enormous amount of sensor data. Accordingly, important new challenges need to be addressed, and a rethinking in the usual computing and networking concepts may be required [4].

Consider a wireless sensor network (WSN) where nodes take sensor readings of the same type (e.g. 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 minimum temperature (MIN) among all nodes at a given moment. Performing such computations at high speed is important because an extreme sensor reading which is correct indicates that the physical process is in an abnormal (and usually hazardous) state which requires that the computer system acts quickly. Achieving such computation quickly also makes it possible to operate nodes at a low duty cycle and thereby save significant amounts of energy.

The problem of data aggregation in WSN has been subject of considerable attention. Techniques have been proposed for computing aggregated quantities such as minimum (MIN) and maximum (MAX) values, the number of nodes (COUNT) and the MEDIAN among a set of sensor nodes. These techniques often involve organizing the nodes in a distribution tree [5], [6] and have leaf nodes broadcast their data. Every non-leaf node waits until it has received a broadcast from its children and then make a single broadcast. Such schemes offer good performance, due to the exploitation of opportunities for parallel transmission, but unfortunately the time-complexity of these approaches depends on the number of sensor nodes.

However, if we envision applications where even a small broadcast domain may contain several tens to a few hundred sensor nodes, the advantages of data aggregation solutions found in previous research are

lost (nodes are in the same broadcast domain if it holds that (i) a wireless broadcast made by one sensor node reaches all other sensor nodes in the broadcast domain and (ii) if a sensor node in the broadcast domain transmits a packet, then it can be received by another sensor node only if the transmission of the packet does not overlap in time with another packet transmission).

We have developed our research around this problem. We want to compute aggregate quantities (such as MIN or MAX) with a time-complexity that is independent of the number of nodes. In fact, we want to compute MIN with a time-complexity that is equivalent to the time of transmitting a single message, even if hundreds of nodes are in the same broadcast domain. Note that, local aggregation between nodes in geographic proximity can be used as an intermediate step to compute aggregated quantities among all nodes in a multihop network; and hence the ability to compute aggregated quantities in a single broadcast domain forms an important building block for many applications.

Consider a simple application as depicted in Figure 1(a), where a node (node N_1) needs to know the MIN of the temperature readings among its neighbors. A naïve approach to this problem would imply that N_1 broadcasts a request to all other nodes and then waits for the corresponding replies from them. Nodes may have set up a scheme to orderly access the medium in a time division multiple access (TDMA) fashion, the initiator node (N_1) knows the number of nodes, and 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 nodes (m).

To illustrate how one can compute MIN with a time-complexity that is equivalent to the time of transmitting a single message, let us now turn to the solution depicted in Figure 1(b). Suppose that the temperature values are coded as n-bit integers. Starting with the most significant bit first, let each node send the temperature reading bit-by-bit. Consider also that, for each transmitted bit, nodes read the resulting value in the channel (something straightforward in a wired medium) and the channel implements a logical AND of the transmitted bits. Furthermore, if a node reads '0' and is transmitting a '1', it stops transmitting. Then at the end of the transmission of n bits, the "observed" value in the channel will correspond to the MIN. It is as if all m temperature readings were transmitted in parallel.

There exist medium access control (MAC) protocols that exhibit this logical AND behavior. This family of protocols is known as Dominance (or, Binary-Countdown) protocols [7]. In the implementations of this protocol (e.g., the Controller Area Network - CAN, currently deployed in hundreds of millions of units [8]), messages have a unique contention field. Typically, the contention field corresponds to a priority that is used to resolve collisions in a non-destructive way – the message with the highest priority (i.e., the lowest value) dominates and thus is transmitted after the collision resolution phase. We use the contention field differently though: during runtime, the contention (or priority) field is computed as a function of the physical quantity of interest. For the simple case of MIN, there is no need for transmitting a message payload, as all information is conveyed in the contention field.

The implementation of a dominance MAC protocol for wireless media in [9] demonstrated that it is also

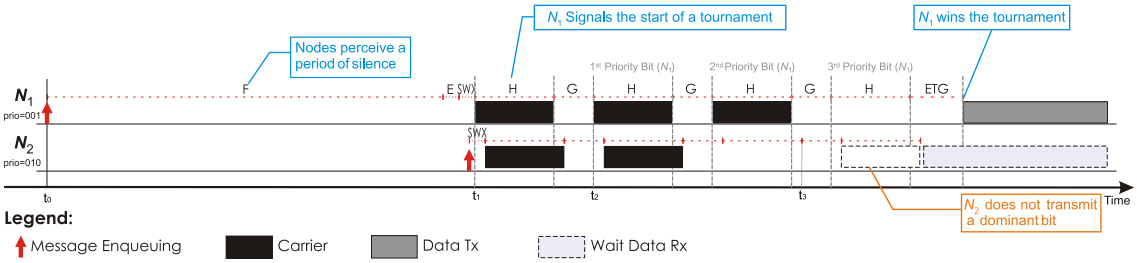


Fig. 2. Dominance-based MAC Protocol.

possible to have Dominance (or, Binary-Countdown) protocols in wireless systems. This implementation was used for real-time communication and never used for data aggregation; in addition, it suffered from a significant overhead because of the time required to (i) perform carrier sensing and (ii) switch between transmit and receive modes. A platform with better such characteristics however has the potential to reduce the overhead and thereby render possible highly scalable aggregate computations for cyber-physical systems.

In this paper, we demonstrate that highly scalable aggregate computations in wireless networks are possible. We do so by (i) building a new wireless hardware platform[1] with appropriate characteristics for making dominance-based MAC protocols efficient, (ii) implementing dominance-based MAC protocols on this platform, (iii) implementing distributed algorithms for aggregate computations (MIN, MAX, Interpolation) using the new implementation of the dominance-based MAC protocol and (iv) performing experiments to prove that such highly scalable aggregate computations in wireless networks are possible.

The remainder of this paper is structured as follows. Section 2 introduces technological background on wireless dominance, including limitations of currently available implementations. Section 3 presents the new platform, its design rationale and evaluation. Section 4 presents the algorithms for aggregate computations, demonstrates experimentally that they work and that they are efficient. In Section 5, related work is discussed. Finally, conclusions are drawn in Section 6.

2 Dominance in Wireless Media

Achieving dominance in the wireless domain is challenging. To begin with, it is not possible to directly translate the behavior of wired protocols, as these require that nodes are able to transmit and receive at the same time. This is not possible in common radio transceivers, because the transmitted energy is much higher than the received energy. For this reason, dominance in wireless systems was previously achieved using a simple principle: when the transmitted bit is dominant, a pulse of a carrier wave is

1. Schematics and source code is available at http://www.hurray.isep.ipp.pt/activities/PRIOMAC/

transmitted and there is no need to sense the medium. Conversely, when the bit to transmit is recessive, nothing has to be effectively sent, instead only the medium state has to be sensed [9].

This approach requires that nodes are able to detect a pulse of the carrier of short duration and it is well known that detection of pulses of short duration is difficult [10]. Moreover, there exist priority levels for which the protocol needs to switch between transmit and receive modes for every priority bit, and this is difficult because most transceivers are not designed for frequent switching and hence every switching takes a non-negligible amount of time.

The challenges of implementing wireless dominance have been addressed by WiDom [9], a protocol that implements Dominance/Binary-Countdown protocols in wireless systems. It was initially proposed assuming that all nodes were within each other's radio coverage – a single wireless broadcast domain. Later, the WiDom protocol was also extended to consider multiple broadcast domains [11].

We will (in Section 2.1) describe the previously known WiDom protocol for single broadcast domain and then (in Section 2.2) discuss why available transceivers are unfit for WiDom and why a new platform is needed.

2.1 WiDom

In WiDom, 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, 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.

Figure 2 presents an exemplifying timeline where three nodes contend for the medium, using 3 priority bits and WiDom operates in a single broadcast domain. The priorities of the messages enqueued at nodes N_1 and N_2 are 1 and 2, respectively. At time t_0, the medium changed from busy to idle, and thus all nodes started detecting a period of silence in the medium. The protocol enforces this period of silence, denoted as F, to guarantee that no node disrupts an ongoing tournament. Notice that nodes can have slightly different perceptions of the evolution of the protocol and the protocol is designed to encompass such factors.

Let us now turn the attention to N_1 in Figure 2. N_1 has a message pending, and after waiting for F time units, it is in position to start a tournament. Before doing so, it must wait for E time units to encompasses possible clock differences between the nodes and to ensure that all nodes have time to listen

for F, after this, the node issues a command to the radio to start sending a carrier wave that will signal the start of a tournament. Because the radio needs to switch from reception to transmission, this carrier wave will effectively be on SWX time units after, at time t_1; SWX represents the time to switch the transceiver from reception to transmission and vice-versa. This carrier wave provides a common reference point in time for all nodes and will be transmitted during H time units, where H is defined to include the time needed to detect a carrier wave (denoted as $TFCS$) and to account for clock imperfections and receive/transmit switching time.

To achieve more accurate synchronization and reduce the overhead, in this paper, we replace this initial behavior. We let a special node send pulses periodically on a separate channel, such that each pulse indicates that a new tournament should start.

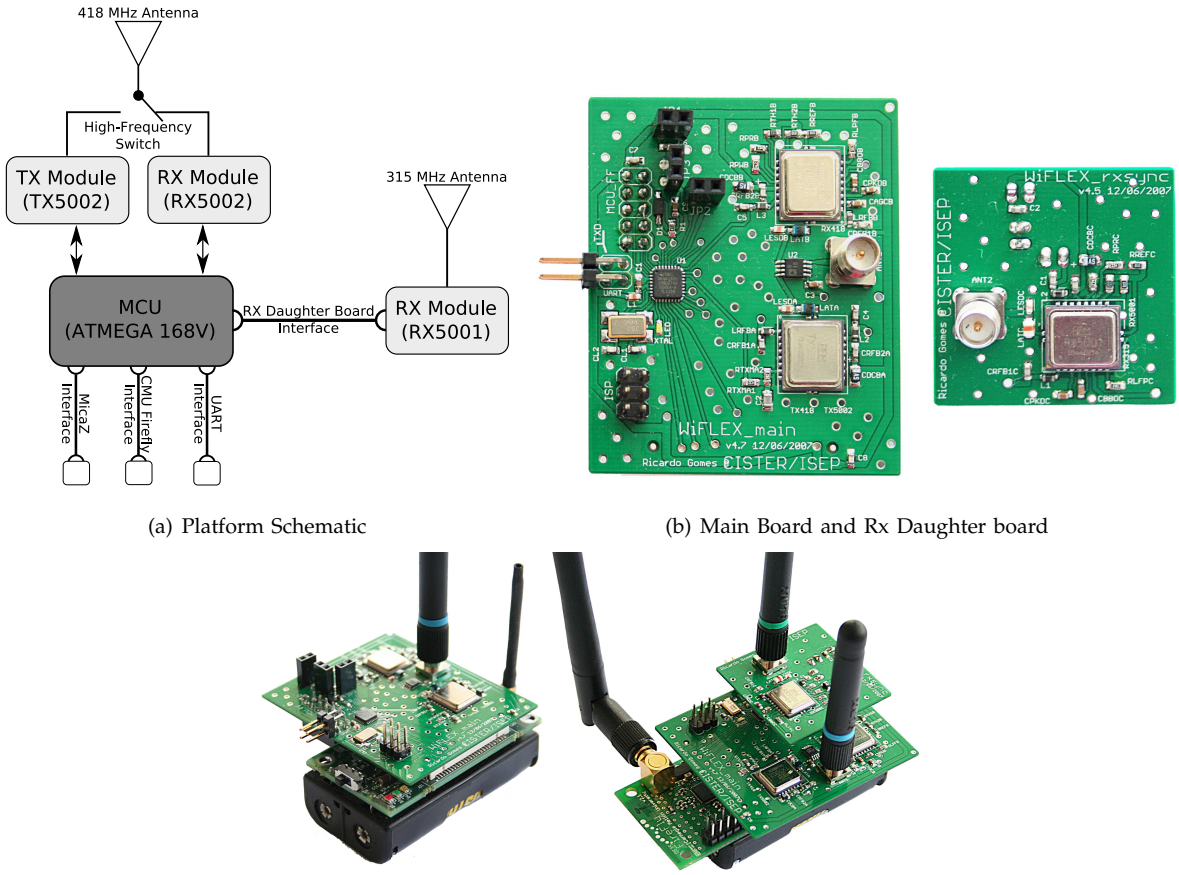
After transmitting the initial carrier wave, N_1 waits for the guarding time interval to separate pulses of carrier waves, denoted by G. This guarding time interval also takes into account possible clock differences. At time t_2, the node starts the tournament, and thus sends its priority bits. Each priority bit is composed of an interval of time where nodes transmit/receive their priorities (H) and a guarding time (G) In Figure 2, N_1 sends a carrier wave during the two first dominant priority bits and then monitors the medium in the last bit, because it is recessive. At the end of sending the priority bits, N_1 never detected a dominant bit from another node, thus it may conclude that it has won the tournament (it will be the only node reaching this conclusion, if priorities are unique). Therefore, N_1 waits for ETG units of time to guarantee (even in presence of clock inaccuracies) that other nodes have time to switch their radios to receive and then starts transmitting the data message.

In the case of N_2, at time t_3 it ends sending a recessive priority bit, thus N_2 monitored the medium for H time units, and detected the carrier wave sent by N_1. This causes node N_2 to refrain from transmitting any further bits, and it starts monitoring the medium until the end of the tournament. Therefore, N_2 will know the priority of the winner. If other nodes existed, they would also follow the development of the tournament and thus also know the priority of the winner.

2.2 Impact of Hardware Shortcomings

While wireless dominance was successfully achieved by WiDom [9], the implementations available are based on off-the-shelf WSN platforms, with a radio transceiver that does not have favorable characteristics for the implementation of a wireless dominance protocol, and thus, these implementations exhibited a considerable overhead. Specifically, the radio transceiver used in [9] was the Chipcon CC2420 [12], a radio transceiver found in many WSN platforms. This transceiver, does not offer the most desirable characteristics for the implementation of dominance protocols. While the specific reasons may vary, this is unfortunately also true for a number of other radios currently used in WSN platforms.

First, WiDom requires that a carrier wave is transmitted for a short duration of time. While some radio transceivers allow to do this (like the CC2420), other radio transceivers only have a byte interface



(a) Platform Schematic

(b) Main Board and Rx Daughter board

(c) Platform Stacked on MicaZ and Firefly

Fig. 3. The New Hardware Platform.

with the microprocessor, which limits the granularity of the duration for the transmission of carriers and introduces unnecessary overhead.

Second, WiDom requires that the radio is able to detect whether other nodes transmit a carrier wave. The ability to detect short pulses of carrier waves is instrumental for the development of an efficient dominance protocol. For example, the CC2420, can detect pulses of carrier waves, using its Clear Channel Assessment (CCA) functionality. The CCA functionality of the CC2420 radio computes the average Receiver Strength Indicator (RSSI) over the last 128 μs. To make a decision, this average is compared to a configurable threshold and then the CC2420 sets the CCA digital output pin accordingly. In practice, this means that $TFCS$ will never be smaller than 128 μs. As an example, in [9], carrier pulses needed to be $TFCS = 486$ μs long in order to be reliably detected (using the transceiver's default threshold).

Finally, it is also necessary that the time to switch between transmission and reception is small. The CC2420, as an example, can take up to 192 μs to switch between these two modes, and then it needs another 128 μs ($SWX = 320$ μs) until the first CCA operation can be made.

The combination of these factors results in wireless dominance that introduces a large overhead, and this limits the usefulness of such implementations in practice.

In Section 3, we will address the issue of designing a platform that allows reducing the SWX and $TFCS$ times, and develop a competitive implementation of the WiDom protocol to enable efficient distributed computations of aggregated quantities in cyber-physical systems.

3 THE NEW PLATFORM

To address the problems described previously, we have developed a platform in the form of an add-on board that can be plugged into common WSN platforms such as the Mica family (Mica, Mica2 and MicaZ) and the CMU-FireFly [13].

3.1 Description

The overhead of the contention of WiDom is dependent on (i) the switching time between transmission and reception mode during the tournament and (ii) the accuracy of the synchronization on the time when nodes should start the tournament.

We ensure a low switching time by using two independent radio modules: one receiver and one transmitter. To allow the use of only one antenna, both modules share a common one, using a high-frequency switch.

We ensure good accuracy of the synchronization on the time when nodes should start the tournament by letting a special node (which we call *master node*) send pulses periodically on a separate channel and when nodes receive a pulse they start executing the tournament. Each node (not the master node) has a separate receiver to detect those pulses. This brings the advantage that this receiver is always in reception mode so the synchronization is very accurate. This solution brings two additional advantages (i) there is no need to wait for F time units as shown in Figure 2 and (ii) the master node can transmit at high transmission power and use a frequency that gives a long range and this makes all nodes synchronized even in a multiple-broadcast network.

Our platform is comprised of a main board and a daughter board. They are constructed such that the main board is attached to a sensor node platform (Mica or FireFly) and the daughter board is attached to the main board. The main board sends and receives pulses in the tournament; the daughter board receives pulses on the separate channel and these pulses indicate the beginning of a tournament.

Figure 3(a) depicts a schematic of the main board and the daughter board. The platform includes interfaces to the Mica family and FireFly sensor platforms, an UART interface, for debugging purposes and an interface to the optional receive-only daughter board. Figure 3(b) shows the platform hardware in closer detail and Figure 3(c) depicts how the new boards stack on common sensor platforms.

The correct operation of WiDom is very sensitive to timing; for example if the transmission of a carrier wave is performed $50\mu s$ later than it should, then the correctness property of WiDom (that the node

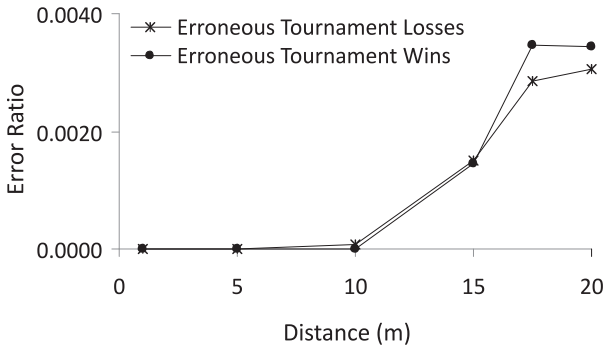


Fig. 4. Failed Tournaments with Distance

with the lowest priority number wins) may be violated. For this reason, the main board is equipped with a microcontroller dedicated for running the WiDom protocol. It controls the radio modules and the high-frequency switch. And it receives commands from the host sensor node platform which priority it should compete on the channel with. It also communicates the priority of the winner to the sensor node platform. It is possible to transmit packets with the TX module on the main board but we choose to not do so because more reliable packet transmission can typically be achieved with the transceiver on the sensor platform.

3.2 Achieving Reliable Tournaments

To achieve reliable tournaments, three main aspects had to be dealt with: symbol encoding, capture effect and bit stuffing. In wireless dominance, dominant bits are transmitted as a pulse of a carrier wave. To account for effects such as a node being able to detect a pulse from a node close by, but because it has adjusted its sensitivity for receiving from this node, it then cannot detect a pulse from a node far away, these pulses must be transmitted with a period of silence between. This allows the receiver to adjust its sensitivity in order to detect carrier waves sent by distant nodes. Note that the switching time from transmit to receive mode is included in this period of silence. The main receiver module becomes unreliable when the medium is idle for a long period of time. This requires two modifications to the original WiDom protocol: (i) the first pulse after a long period of silence is composed of several pulses to adjust the receiver into an active state and (ii) bit stuffing must be introduced during the tournament, to avoid long periods of silence due to several consecutive recessive bits.

The length of the periods of silence between pulses and the amount of bit stuffing necessary is determined through experimentation and is described in Section 3.3.

3.3 Evaluation

We are interested in investigating the following characteristics of our platform: (i) finding the minimum pulse duration and the transmission/reception switching time; (ii) how the reliability changes with distance; (iii) determining the power consumption.

To this end, several experiments were carried out. All experiments were conducted in an open-field environment, and all nodes were in non-obstructed line-of-sight.

First, to determine the minimum pulse duration and the transmission/reception switching time, we set two nodes at a distance d apart, and increased d in steps of 1 m, starting from 5 meters. For each step, we used the daughter board to perform synchronization using an out-of-band signal and performed 10000 tournaments using a static priority in each node. This procedure was then repeated for different combinations of pulse widths and intervals of silence. As a result of this experiment, we selected a pulse width of $H = 40\ \mu s$ and an interval of silence between the bits $G = 50\ \mu s$, which was the combination that performed best. Using these values, and considering bit stuffing, we get a tournament duration of $2700\ \mu s$ for 15 priority bits.

To exhaustively test the reliability of the tournament at different distances, we performed an experiment with 10 nodes. Furthermore, to test for cases where nodes can detect pulses from a node close by, but not from a node far away, we divided the nodes into two groups and placed one group of nodes at each end. The nodes in each group were placed with a minimum distance of 30 cm and the distance between the two groups of nodes varied from 1 to 20 meters, in steps of 5 meters. We produced a table of random priorities and computed the winner of each tournament offline. This table was then downloaded to the nodes and the priorities in the table were used in cycles until 150000 tournaments were performed. Because nodes know the priority of the winner in each tournament from the table computed offline, the number of tournaments failed (both events of failing to win or lose a tournament were counted) could be collected from each individual node.

The results are presented in Figure 4. It is possible to observe that, for networks where the distance between all nodes is below 5 meters, the communication can be considered error-free. Also, the number of tournaments failed at a distance of 20 meters is very small.

We also studied the amount of energy consumed by the board. For this, the power consumed by a MicaZ platform with the board and the daughter board attached during a tournament was examined. During the measurements, the radio onboard the MicaZ was switched off, and the microcontroller was performing a busy loop. Because our platform needs to perform bit stuffing, each bit in our trace is composed of the actual bit (a pulse of the carrier wave, or monitor the medium) and an extra carrier pulse as bit stuffing.

We also measured the amount of power that an implementation of wireless dominance using the CC2420 onboard the MicaZ to transmit a dominant and a recessive bit. For our measurements, we used a pulse width of $128 + 128\ \mu s$. This duration was used because it is the minimum time needed to

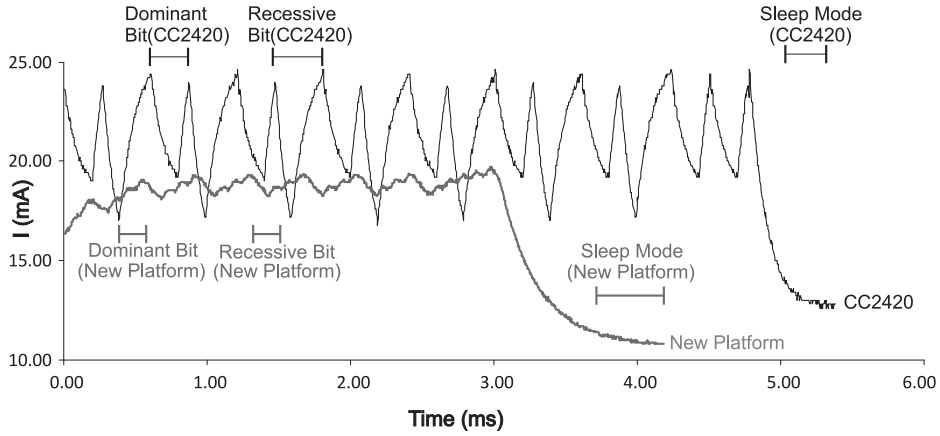


Fig. 5. Trace of Power Consumption

switch to transmit mode and perform one RSSI reading in the CC2420, meaning that carrier sensing will never be faster and thus pulses should be, at least, this big. For the recessive bit, we considered an interval of $192 + 128$ μs, as this is the minimum time to switch to receive mode and again do one RSSI reading. The microcontroller was also performing a busy loop during the measurements. Note that a real implementation (such as the one in [9]) would not be able to use pulses with such short duration to achieve reliable tournaments in a range of several meters.

Figure 5 presents both traces of power consumed (the supply voltage is 3V) for transmitting the same number of priority bits in a tournament. One can observe that the new platform consumes significantly less energy than an implementation using the CC2420, even using the minimum time for the duration of the bits, as described above.

4 The Algorithms

In this section we will discuss several algorithms that are enabled by the efficient implementation of a Dominance MAC protocol for wireless channels, as presented in the previous section. Firstly, we present the algorithms designed for the case of a single broadcast domain (SBD), that is, when all nodes can hear each other. Then, we address the case where we can have multiple broadcast domains (MBD).

4.1 Single Broadcast Domain (SBD)

4.1.1 Computing MIN

We can exploit a Dominance MAC protocol to compute MIN as discussed earlier in Section 1. In such case, the time to compute MIN is equal to the time to perform one arbitration of the dominance MAC protocol.

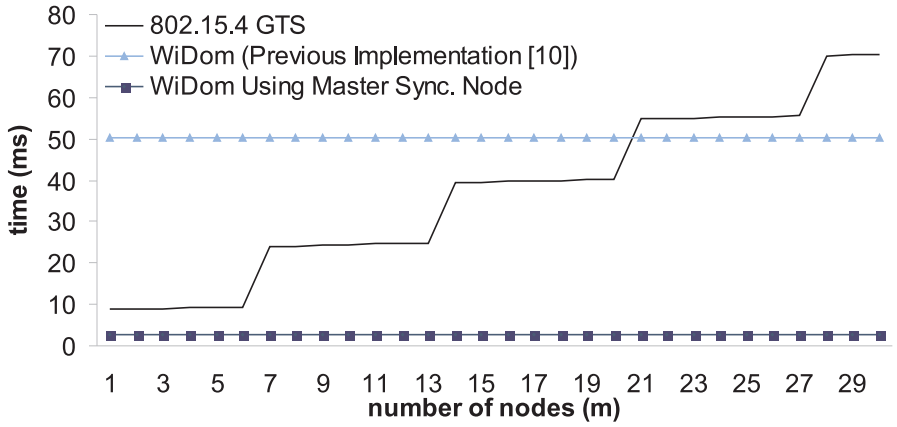
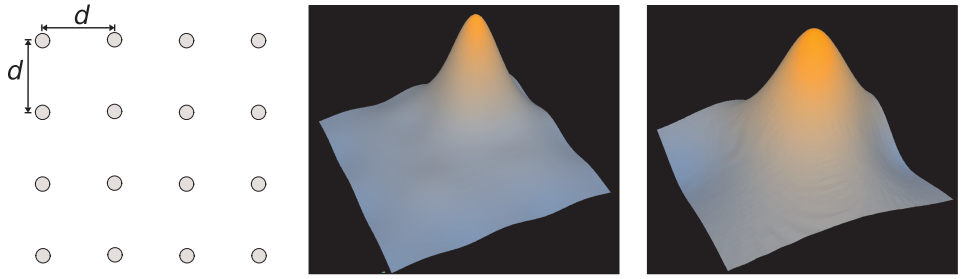


Fig. 6. Time to Compute MIN.



(a) Node Placement Grid (b) Original Signal (c) Resulting Signal ($k = 5$)

Fig. 7. Interpolation Experiment.

Figure 6 shows, for various implementations, the time required to compute MIN as a function of the number of nodes. Two implementations of WiDom and one implementation using IEEE 802.15.4 Guaranteed Time Slots (GTS) are presented. Note that (i) the time to compute MIN using WiDom is independent on the number of nodes and (ii) the new implementation of WiDom is faster than IEEE 802.15.4 GTS for any system size. Further details are available in [14].

4.1.2 Computing MAX

MAX can be computed in an analogous way to MIN. Each node uses the bitwise negation of its sensor reading as a priority and competes for the channel. The MAX is then obtained as the bitwise negation of the winning priority.

4.1.3 Interpolation

A wired dominance MAC protocol has been exploited to track how a physical quantity varies over an area observed by a dense deployment of sensor nodes. This approach, being based on the previously

presented approach to compute MIN/MAX, has a time-complexity that does not depend on the number of sensor nodes, and presents itself as a very appealing approach that tightly couples communications and computations with the physical environment. The new platform presented in the previous sections enables this approach over wireless systems, opening the possibility of deploying large-scale wireless sensor networks that can efficiently track how a physical quantity varies over space. The basis for the design of this approach is an interpolation scheme. This interpolation is a compact representation of sensor data at a given moment and it can be obtained efficiently. This was proposed, implemented and tested for a wired system in [15]. The scheme is as follows.

Each nodes is assumed to know its location. All nodes use the same function to interpolate the sensor data. Nodes start with the interpolation function being a flat surface. Then each node computes the error between its sensor reading and the interpolation function evaluated on its location. Exploiting the dominance MAC protocol, the nodes then use their error as part of the priority used to contend for the medium, such that the data point with the MAX of all errors is found. The data point found is then used by all nodes to recompute the interpolation function. Nodes iterate through this procedure for a predefined number of times (k). At the end of these iterations, the subset of k nodes that contribute to the interpolation is found. k is a parameter that defines the accuracy of the interpolation.

The reasoning for finding only a subset of nodes is explained by the fact that sensor readings often exhibit spatial locality [16]; that is, nodes that are close in space give similar sensor readings. For this reason, the interpolation offers a low error even if only a small number of carefully selected sensors reading is used. Note that not just any subset of nodes is selected; it is the subset of nodes that has, at each iteration, the largest error to the interpolation function, and thus can contribute the most to approximate the physical phenomenon. From the description above, one can see that nodes must broadcast their location after contending for the medium. This requires that the priority used for the contention is divided in two parts: one used for the value computed as a function of the physical quantity of interest, and the other part for a unique identifier of the node. In this way, we guarantee that only one node can win the contention for the medium and no collisions occur during the transmission of the location data.

Implementation and Experiments

To demonstrate and evaluate the interpolation scheme using wireless sensor platforms, we used a MicaZ sensor platform as the host of our add-on board described in Section 3. The algorithm for the interpolation implemented is as presented in [15].

Using this implementation, we have setup an experiment to study the reliability of the interpolation. We placed 16 nodes (the number of prototype platforms available at this time) at equal distances d between each other, to form a 4×4 grid, as depicted in Figure 7(a). We constructed a signal as in Figure 7(b) and fixed the data points in each node according to it.

In this way, the resulting interpolation (depicted in Figure 7(c)) was known in advance. We performed

d (*meters*)	Error Ratio (%)
0.5	0.106
1	0.148
2	0.155

TABLE 1
Interpolation Experiment Results.

the interpolation 10000 times for each time we changed the distance d. The results are presented in Table 1. The experiments were conducted in an office environment.

It is possible to observe that even with such demanding application (performing interpolation requires that not only the node with highest priority wins, but also that all other nodes perceive the right priority of the winner), our platform offers reliable support for dominance-based aggregate computations.

4.2 Multiple Broadcast Domains (MBD)

4.2.1 MIN

The schemes discussed in the previous section are designed for a single broadcast domain. However, often one has to consider networks where this is not the case. In this section, we will extend the scheme to deal with multihop networks.

In order to present this extension, we must layout a few assumptions. The communication range (R_{co}) is the maximum range at which two nodes N_i and N_j can communicate reliably. The interference range (R_{it}) is the maximum range between nodes N_j and N_k such that simultaneous transmissions to N_j will collide with N_k. We assume that $R_{it} \leq 2R_{co}$.

Our scheme 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.

Setup. The setup procedure partitions 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. This is achieved by approximating the solution to a *Minimum Virtual Dominating Set* (MVDS). 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 approximate the solution to a MVDS(r)

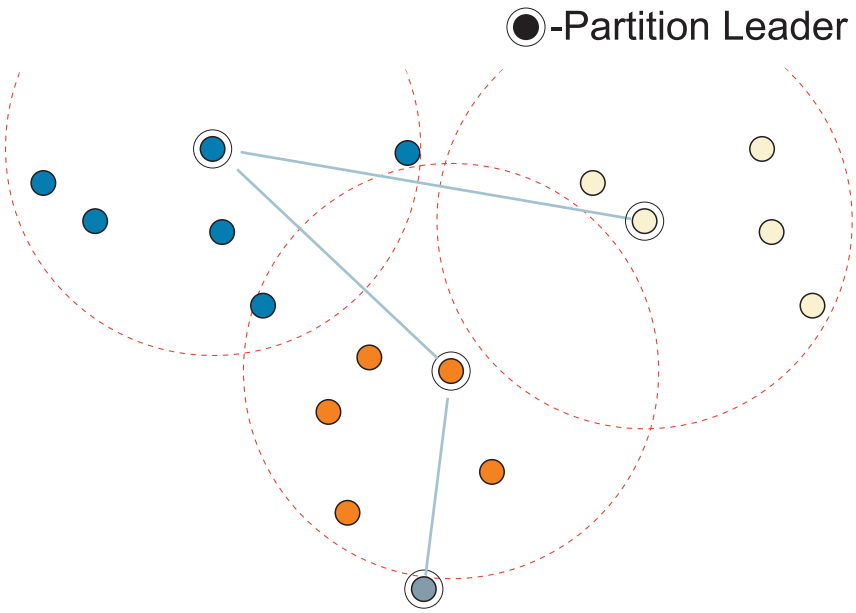


Fig. 8. MVDS Algorithm Illustration

where r is the virtual range defined 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 8. It presents the different partitions formed by representing the nodes in the same partition similarly (each partition has a different symbol and color). The partition leaders selected by the algorithm (identified with a small circle around them) and their respective virtual ranges are also represented (dashed circular red line around partition leaders; remember that the virtual range is $R_{co}/2$). This algorithm to construct the MVDS is based in [17].

After running the propagation phase of the MVDS construction algorithm, the nodes selected as partition leaders report back to the leader the information about the topology of the network. This topology information is used by the leader to assign a timeslot to each partition such that no neighbor nodes or 2-hop neighbor nodes have the same timeslot. The timeslot assigned to each partition is then transmitted to all nodes. The number of timeslots assigned during this step is denoted nts.

Runtime. At runtime, nodes have to find the set of minimum sensor readings within each partition $PART_j$, and then the partition leaders deliver each minimum value to the leader. Algorithm 1 provides the sequence of steps nodes take during runtime.

4.2.2 Computing MAX

As before, MAX can be computed similarly to MIN, using the bitwise negation of sensor readings.

4.2.3 Interpolation

There are two ways of obtaining an interpolation of sensor readings in a multiple broadcast network. One approach is to create partitions like in the MIN calculations and run the algorithm for obtaining the interpolation in each partition. And then let each partition-leader communicate the data points, that constitute the interpolation in that partition, to the leader node of the network.

Another approach start with a flat surface as an interpolation of the sensor signal in the area of the entire network. Then we select the node with the maximum error in the entire area; this can be achieved by computing MAX in a multiple broadcast domain network as suggested in Section 4.2.2. When the leader node knows the node with the maximum error, it propagates this information to all nodes in the network (by sending this information to all partition leader and then let each partition leader broadcast this information). In this way, all nodes in the entire network maintains a set of sensor readings on which they can compute a new interpolation. The procedure is then repeated k times.

5 Related Work

The wireless sensor network (WSN) community has provided four important ideas for aggregated computations. One is to organize nodes as a convergecast-tree [18] such that leaf-nodes broadcast their data; each non-leaf-node waits until all its children have broadcast their data and then it aggregates the sensor readings (for example taking the minimum of the sensor readings provided by its children) and then broadcasts that aggregate. Unfortunately, this idea loses its efficiency when the broadcast domain is very dense.

A second idea is to perform aggregate computations locally in a cluster of nodes (for example, [19]) and then perform aggregated computations among the cluster heads. Those works also suffer from the drawback that in a single broadcast domain, the time-complexity is O(m). A third idea used in the WSN community is to let random variables in the CSMA/CA protocol be a function of the data payload of the packet that is requested to be transmitted. This will promote packets with important data to be transmitted before less important data. Because of the use of randomization, it is difficult to prove upper bounds on the delay. A fourth idea is to let nodes suppress its transmission [20], [21] if it knows that its data payload will not affect or have very small impact on the result of the aggregate computation. For example, consider a node N_1 with temperature 80 degrees and the goal is to find the minimum temperature in an area. If another node has already broadcast a packet stating that its temperature is 78 degrees then N_1 should suppress its transmission. Common to these ideas is that they can reduce the number of messages but the time complexity of computing aggregated computations is still dependent on the number of nodes. And hence large, dense networks in a single broadcast domain will not be able to perform aggregated quantities quickly with those approaches.

Ringwald and Römer [22] created a distributed algorithm (called BitMAC) which allows nodes to organize into a finite number of sets and query if a set is non-empty. The idea is that nodes agree on

Algorithm 1 Computing MIN in MBD

1: Each node N_i in $PART_j$ waits until the timeslot SLOT($PART_j$).
2: All N_j nodes execute the MIN algorithm to compute $SMIN_j$.
3: The partition leader of $PART_j$ communicates $SMIN_j$ to the leader.
4: The leader computes the MIN using all value received.

time intervals, one for each set. If a node N_i is a member of set S_k then node N_i knows that set S_k is non-empty and node N_i broadcasts an unmodulated carrier wave in the time interval belonging to the set S_k. If a node N_i is not a member of set S_k then node N_i performs carrier sensing. If a carrier was detected then N_i knows that set S_k is non-empty; otherwise N_i knows that set S_k is empty. The time complexity of querying is dependent of the number of sets. But the algorithm has the prominent feature that the time-complexity of querying is independent of the number of nodes and in that respect it shares a property with the schemes we propose in this paper. They did not mention computing MIN, MAX or interpolations. And it was noted that the long time required for carrier sensing and switching from TX to RX mode of the transceiver (Chipcon's CC2420 [12]) hampers the performance of their scheme. Therefore their scheme would benefit from using the platform presented in this paper.

We conclude that (i) the current state-of-art does not offer a platform specifically designed for scalable data aggregation and (ii) the current state-of-art offers no demonstration that the aggregated computations based on a prioritized MAC protocol is feasible in the wireless domain. Our work solves these two issues.

6 Conclusions

Computer systems with tight interaction with the physical dynamics must obtain an accurate and timely estimate of the state of physical processes. The former calls for a large number of sensors. The latter calls for algorithms that can perform operations on those sensor readings quickly; it must at least keep the pace with the physical dynamics.

We have seen that such scalable computations are possible even in the wireless domain. Our interpolation technique makes it possible for all sensor nodes to obtain a snapshot of all sensor readings. Although only an approximate representation, it brings exciting opportunities. Virtually any sensor fusion algorithm can now be performed based on that representation.

Acknowledgment

This work was partially supported by the Portuguese Science and Technology Foundation (Fundação para Ciência e Tecnologia - FCT), the ARTIST2 Network of Excellence on Embedded Systems Design, funded by the European Commission under FP6 with contract number IST-004527 and CONET, the Cooperating Objects Network of Excellence, funded by the European Commission under FP7 with contract number FP7-2007-2-224053

REFERENCES

[1] J. A. Stankovic, I. Lee, A. Mok, and R. Rajkumar, "Opportunities and obligations for physical computing systems," *IEEE Computer, 38(11)*, pp. 23–31, November 2005.

[2] D. Estrin, D. Culler, K. Pister, and G. Sukhatme, "Connecting the physical world with pervasive networks," *IEEE Pervasive Computing*, pp. 59–69, January-March 2002.

[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)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 102–108.

[4] E. A. Lee, "Cyber-physical systems - are computing foundations adequate?" in *NSF Workshop On Cyber-Physical Systems:Research Motivation, Techniques and Roadmap (Position Paper)*, 2007.

[5] 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. [Online]. Available: http://www-db.cs.wisc.edu/cidr/cidr2003/index.html

[6] 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.

[7] A. K. Mok and S. Ward, "Distributed broadcast channel access," *Computer Networks*, vol. 3, pp. 327–335, 1979.

[8] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, "Controller area network (can) schedulability analysis: Refuted, revisited and revised," *Real-Time Systems*, vol. 35, pp. 239–272, 2007.

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

[10] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part ii - the hidden terminal problem in carrier sense multiple-access and the busy-tone solution," *IEEE Transactions on Communication*, vol. 23, pp. 1417–1433, 1975.

[11] N. Pereira, B. Andersson, E. Tovar, and A. Rowe, "Static-priority scheduling over wireless networks with multiple broadcast domains," in *Proceedings of the 28th Real Time Systems Symposium (RTSS'07)*, Tucson, U.S.A., December 2007.

[12] Chipcon., "CC2420 datasheet," http://www.chipcon.com/files/CC2420_Data_Sheet_1_3.pdf.

[13] R. Mangharam, A. Rowe, and R. Rajkumar, "Firefly: a cross-layer platform for real-time embedded wireless networks," *Real-Time Syst.*, vol. 37, no. 3, pp. 183–231, 2007.

[14] N. Pereira and B. Andersson, "Widom vs ieee 802.15.4 for computing min in a single broadcast domain," IPP Hurray! Technical Report HURRAY-TR-081003, online at http://www.hurray.isep.ipp.pt/private/HURRAYTR081003.pdf, 2008.

[15] 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*, vol. 4, May, 2008, pp. 80–91.

[16] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, "Distributed regression: an efficient framework for modeling sensor network data," in *Proceedings of the Third International Conference on Information Processing in Sensor Networks (IPSN04)*, 2004.

[17] B. Deb, S. Bhatnagar, and B. Nath, "Multi-resolution state retrieval in sensor networks," in *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, 2003, pp. 19–29.

[18] A. Skordylis, N. Trigoni, and A. Guitton, "A study of approximate data management techniques for sensor networks," in *Proceedings of the fourth Workshop on Intelligent Solutions in Embedded Systems WISES'06*, 2006.

[19] P. Popovski, F. H. P. Fitzek, H. Yomo, T. K. Madsen, and R. Prasad, "MAC-layer approach for cluster-based aggregation in sensor networks," in *Proceedings of the International Workshop on Wireless Ad-hoc Networks*, Oulu, Finland, May–Jun. 2004, pp. 89–93.

[20] K. Jamieson, H. Balakrishnan, and Y. C. Tay, "Sift: a MAC protocol for event-driven wireless sensor networks," in *Proceedings of the third European Workshop on Wireless Sensor Networks (EWSN'06)*, 2006, pp. 260–275.

[21] M. C. Vuran and I. F. Akyildiz, "Spatial correlation-based collaborative medium access control in wireless sensor networks," *IEEE/ACM Transactions on Networks*, vol. 14, no. 2, pp. 316–329, Apr. 2006.

[22] M. Ringwald and K. Römer, "BitMAC: A deterministic, collision-free, and robust MAC protocol for sensor networks," in *Proceedings of 2nd European Workshop on Wireless Sensor Networks (EWSN 2005)*, Istanbul, Turkey, Jan. 2005, pp. 57–69.