Highly Scalable Aggregate Computations in Cyber-Physical Systems: Physical Environment Meets Communication Protocols

Eduardo Tovar, Björn Andersson, Nuno Pereira, Mário Alves,
Shashi Prabh and Filipe Pacheco
IPP-HURRAY Research Group, CISTER/ISEP, Polytechnic Institute of Porto, Portugal
{emt, bandersson}@dei.isep.ipp.pt, {nap, mjf, ksph,ffp}@isep.ipp.pt

Abstract

In this paper, we focus on large-scale and dense Cyber-Physical Systems, and discuss methods that tightly integrate communication and computing with the underlying physical environment. We present Physical Dynamic Priority Dominance ((PD)^2) protocol that exemplifies a key mechanism to devise low time-complexity communication protocols for large-scale networked sensor systems. We show that using this mechanism, one can compute aggregate quantities such as the maximum or minimum of sensor readings in a time-complexity that is equivalent to essentially one message exchange. We also illustrate the use of this mechanism in a more complex task of computing the interpolation of smooth as well as non-smooth sensor data in very low time-complexity.

1. Motivation

Although the Information Technology (IT) transformation of the 20th century appeared revolutionary, a bigger change is on the horizon. The term Cyber-Physical Systems (CPS) [1] has come to describe the research and technological effort that will ultimately allow interlinking of the real-world physical objects and the cyberspace efficiently. A few other terms have been used to describe similar endeavors. The term “Internet of Things,” originally aimed at RFID-related technologies [2], is gradually becoming a synonym for Cyber-Physical Systems.

The integration of physical processes and computing is not new. Embedded systems have been in place for a long time and these systems often combine physical processes with computing. The revolution will come from massively networked embedded computing devices, which will allow instrumenting the physical world with pervasive networks of sensor-rich embedded computation [3].

With the march of Moore’s law, size and cost of sensor nodes continue to decrease, thus enabling the implementation of systems with increasingly larger number of nodes. Recently, networks with more than one thousand sensor nodes [4] have been deployed for collaborative processing of physical information. It is expected that networks with hundreds of thousands of nodes will be deployed within a few years from now, thus realizing Mark Weiser’s vision [5]. In the long-term, one can expect networks with millions of sensor nodes in operation. Such large-scale sensor-rich networked systems will generate an enormous amount of sensor data. Accordingly, important new challenges need to be addressed. Such systems require rethinking of the usual computing and networking concepts [6]. Furthermore, given that the computing entities interact with their environment, often timeliness is of paramount importance.

To illustrate this vision, consider a large-scale dense networked sensor system whose nodes have a common sensing goal to measure temperature. Now consider the problem of computing a simple aggregate quantity: the minimum (MIN) sensed temperature among the nodes at some given moment. Computing MIN seems trivial, but for systems such as those described above, it poses an important problem – communicating sensor data individually makes the time-complexity of computing MIN a function of the number of nodes. This is true even if data aggregation is used.

In multihop networks, nodes may self-organize into a convergecast tree with a base station at the root. Techniques for computing useful aggregated quantities such as MIN that offer good performance have been proposed previously [7, 8]. Such techniques for the convergecast topology achieve good performance as a result of exploiting the opportunities for parallel transmission and of en-route aggregation of data.

Unfortunately, these advantages are lost when all nodes share a single broadcast domain. In a single broadcast domain (wired as well as wireless), it holds that: (i) a broadcast made by one sensor node reaches all other sensor nodes; and (ii) if a sensor node transmits a message, then it can be received by another sensor node only if the transmission of the message does not overlap in time with another message transmission.

Even a small broadcast domain (covering an area smaller than 10 m^2) may contain a few hundred sensor nodes [9]. Furthermore, 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 solution to the problem of computing aggregated quantities in a single broadcast domain forms an important building block for many wireless (or wired) sensor network applications [10].

2. The (PD)^2 Protocol

We have an ambition: being able to compute MIN (or MAX) with a time-complexity that is independent of the number of sensor 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 thousands of nodes are in the same broadcast domain. *Is this possible*? In this paper, we provide supporting evidence that the answer is in the affirmative.

Assume a networked sensor system with m nodes where each node has an n-bit temperature sensor. Computing MIN implies that all m individual values are compared. Ordinarily, it will take $O(m)$ message transmissions. Furthermore, due to packet collisions, we can not assume to transmit all m messages simultaneously.

For the simplicity sake, we assume that the temperature values are coded as n-bit unique integers. Starting with the most significant bit first, let each node send the temperature reading bit-by-bit. Let us consider that the channel implements a logical AND of the transmitted bits and for each transmitted bit and nodes read the resulting AND value in the channel (something straightforward in the wired medium). Finally, suppose that 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. Observe that for this case of computing MIN, there is no need for message payload. Observe that the responses are generated only as the result of some query that is received by all nodes of the broadcast domain. Therefore, it is not required that all clocks agree on a common time.

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 [11]. In the implementations of this protocol (e.g., the Controller Area Network or, CAN), messages have a unique contention field, which typically corresponds to a priority that is used to resolve the contention for channel access. After the completion of contention resolution phase, the node having message with the highest priority is granted channel access.

We propose to use the contention field differently: during runtime, the contention (or priority) field is computed as a function of the physical quantity of interest. We denote this simple, but powerful, mechanism as Physical Dynamic Priority Dominance ((PD)^2) protocol. We advocate its use as a key component in sensor applications where it is crucial to compute aggregate quantities with low time-complexity, even for very dense systems. The (PD)^2 protocol is in fact an example where communication and computation are tightly connected with the physical environment, which is a fundamental feature of CPS.

Besides MIN, it has been shown that (PD)^2-like mechanism can be used to compute interesting primitives such as the maximum of sensor readings (MAX), an estimation of number of nodes (COUNT) and an estimate of the median of sensor readings (MEDIAN) [12].

MAX can be obtained by computing the MIN of the difference between a number that is larger than the largest possible sensor reading and the sensor readings. The intuition behind our method of estimation of COUNT is as follows: If the contention field is a non-negative random number obtained at runtime, then the probability that the minimum value of the contention field is 0 approaches 1 as the number of nodes get very large. However, if there are only few nodes, then it is highly unlikely that the minimum among the random values is zero. From this observation, one can see that it is possible to estimate the number of nodes by computing the MIN of the random numbers. For more details on COUNT, please see [12]. In this case, MIN is not a function of a sensed physical quantity, instead it is a function of random variables which are used to estimate a physical quantity. COUNT can then be used as a basic building block to estimate MEDIAN. A panoply of functions may eventually be devised out of protocols similar to (PD)^2 since any logical function can be implemented in terms of the NAND or NOR primitives.

3. Related Work

The (PD)^2 protocol is inspired by Dominance protocol [11] that was implemented for wired networks in the widely used CAN bus [13]. In [12], the authors illustrate that CAN-enabled platforms can be used to compute various aggregate quantities using a (PD)^2-like mechanism.

WiDom protocol extends the Dominance protocols to wireless networks consisting of single broadcast domain [14]. Wireless transceivers do not transmit and receive data simultaneously. Therefore, for wireless networks, a node "transmitting a 1" performs carrier sensing only. All nodes "transmitting a 0" transmit simultaneously. Thus, those nodes that were “transmitting a 1” (i.e., doing carrier sensing) and sense a ‘0’ being transmitted stop contending for channel access any further. Given the growing importance of wireless sensor networks, this extension is significant. Later, that work was generalized to multiple broadcast domains [15]. It is important to note, however, that the current implementations of WiDom introduce a significant overhead. To a large extent, this overhead is due to large switching time of transceiver’s transmission and reception modes and to the time needed to perform carrier sensing. This is, nevertheless, a technological limitation that can be overcome with adequate hardware (see Section 5.3 in [15] for a discussion on this issue).

WiDom has also been applied to compute aggregate values of sensor data in multi-hop wireless sensor networks [16]. In this case, the algorithm exhibits a time complexity that depends on the network diameter and on the range of sensor reading values.

In a recent research, (PD)^2-based mechanism has been applied to compute the interpolation of sensor data [12]. This system has also been implemented on CAN-enabled platforms. The interpolation algorithm of [12] performs well for smooth sensor data (or, signals). However, its performance on non-smooth sensor data degrades. In the following section, we present an interpolation method for smooth as well as non-smooth sensor data and sketch a (PD)^2-based solution for it.

4. Interpolation with (PD)^2

Consider a WSN deployment monitoring physical entities such as temperature, humidity, noise, ambient light etc. In this scenario, one may want to obtain an interpolated map of these entities over the area of deployment. A subset of data values combined with their sampling location, henceforth referred to as data points, is needed to obtain the interpolation. Smaller subset size translates to smaller cost to compute the interpolation map. In the following, we consider obtaining the interpolation using (PD)^2 protocol in a single broadcast domain. This assumption implies that *every* node of the domain can compute the interpolated map.

A random selection of data points leads to poor interpolated map (Figure 2d). Suppose that an approximate interpolation map is given. Then intuitively, a good strategy for iteratively improving the accuracy of this map is to include the data point that differs the most from the approximate map followed by re-computation of the map. In the following, at each iteration t the (PD)^2 protocol compares actual data with the interpolated values of step $t-1$ *locally* and augments the interpolation data point subset with data from one of those locations that have the largest deviation. Simulation shows that such a selection of data points produces surprisingly good interpolation map.

Similar to the work presented in [12], we use weighted-average interpolation [17] as the interpolation function. However, we introduce a new criterion to select the data points that improves the performance of our interpolation scheme for non-smooth signals as well. The details of this interpolation scheme can be found in [18]. We present an overview of the solution next.

4.1 Interpolation Algorithm

Consider a sensor network where each node has a unique identifier. Let us denote the location of node N_i by (x_i, y_i) and its sensor reading by s_i. Our solution computes the interpolation map iteratively, and the number of iteration steps, k, is assumed to be known to all nodes. Let Q denote a set of tuples. Each tuple $q_i \in Q$ is described by (x_i, y_i, s_i), which corresponds to the location and sensor reading of node N_i. At the beginning of the iteration, the set Q is empty. At each step of the iteration, one of the nodes broadcast its location and sensor data value, which is added to the set Q. Eventually, the set Q contains k elements. We use a subset, $T(x,y)$, of Q to compute the interpolation at (x,y). First, we describe the algorithm to construct Q. Then we follow up with the description of the algorithm to construct $T(x,y)$ from Q.

We define $f(x,y)$, the function that interpolates the sensor data, as follows:

$$f(x,y)=\begin{cases} 0 & \text{if } Q=\emptyset; \\ s_i & \text{if } \exists q_i \in Q\text{: } x_i=x \wedge y_i=y; \\ \dfrac{\sum\limits_{\forall q_i \in T(x,y)} s_i \times w_i(x,y)}{\sum\limits_{\forall q_i \in T(x,y)} w_i(x,y)} & \text{otherwise,} \end{cases} \tag{1}$$

where the weights $w_i(x,y)$ are given by:

$$w_i(x,y) = \frac{1}{(x_i-x)^2+(y_i-y)^2}. \tag{2}$$

The first case of Equation 1 represents the initialization. The second case states that if the exact data at some location has been communicated, then the exact value is used as the interpolated value. The third case states that the interpolated value is a weighted average and the weight is the inverse of the square of the distance.

We construct the set Q by successively adding tuples of those nodes N_i that have the largest difference between the interpolated sensor reading $f(x_i, y_i)$ and the actual sensor reading s_i. Formally, let the magnitude of the interpolation “error” at node N_i be defined as:

$$e_i \triangleq |s_i - f(x_i, y_i)|. \tag{3}$$

The node with the largest e_i is selected for broadcasting its data point. This reduces the problem of selection of most

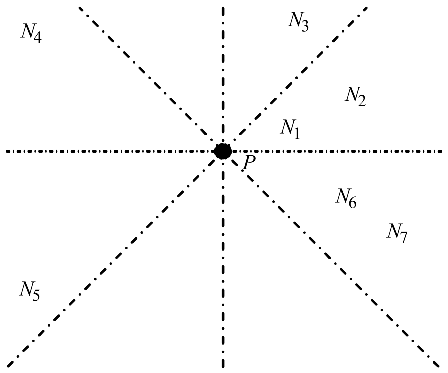


Figure 1. Interpolation at a point P

suitable data point for interpolation to the problem of finding MAX. We discussed obtaining MAX using the (PD)^2 protocol in Section 2. Observe that the winning node must also broadcast its position. In order to broadcast a packet after winning the channel access without collision, we must ensure a unique winner. We achieve this by appending the node ID to the contention field. Upon k broadcasts of data points the construction of the set Q completes.

Now, we describe construction of set $T(x, y)$, which is used to compute the interpolation at location $P(x, y)$ using Equation 1. Consider the region containing nodes N_1 and N_2 in Figure 1. To obtain interpolation at P, we consider N_1's data only – irrespective of the difference in the data values at N_1 and N_2. Intuitively, if the difference is significant, for example due to a wall in between the two nodes, then taking N_2's data into account will lead to interpolation inaccuracy. In other words, N_1's data "masks" N_2's. Following this argument, we use the nearest neighbor of P from each of the sectors to construct $T(x, y)$. Therefore, in this example, $T(x, y) = \{N_1, N_3, N_4, N_5, N_6\}$. More details can be found in [18].

4.2 Simulation Experiments

We evaluated the performance of our interpolation solution using simulation (We implemented our own simulator in C.) We generated a bathtub looking non-smooth sensor data distributed over 1000X1000 grid (Figure 2a). It may be noted that this data represents a general distribution of heat in a room, albeit inverted [12]. We added uniformly distributed random noise to this data distribution (Figure 2b) and we used the resulting data with noise for the evaluation of our algorithm.

Figure 2c shows the interpolation map obtained after only 20 iterations. The data points selected in the iterations on the basis of largest error by (PD)^2 protocol are marked with a bar parallel to the z-axis. In Figure 2d, we present an interpolated map based on random sampling of data points. One can easily see the merit of our algorithm by comparing these two figures. Figure 2e shows the result of sampling 1000 data points, which is only 0.1% of the input size. Even at such a small sample fraction, the reproduction of the original data distribution becomes remarkably accurate.

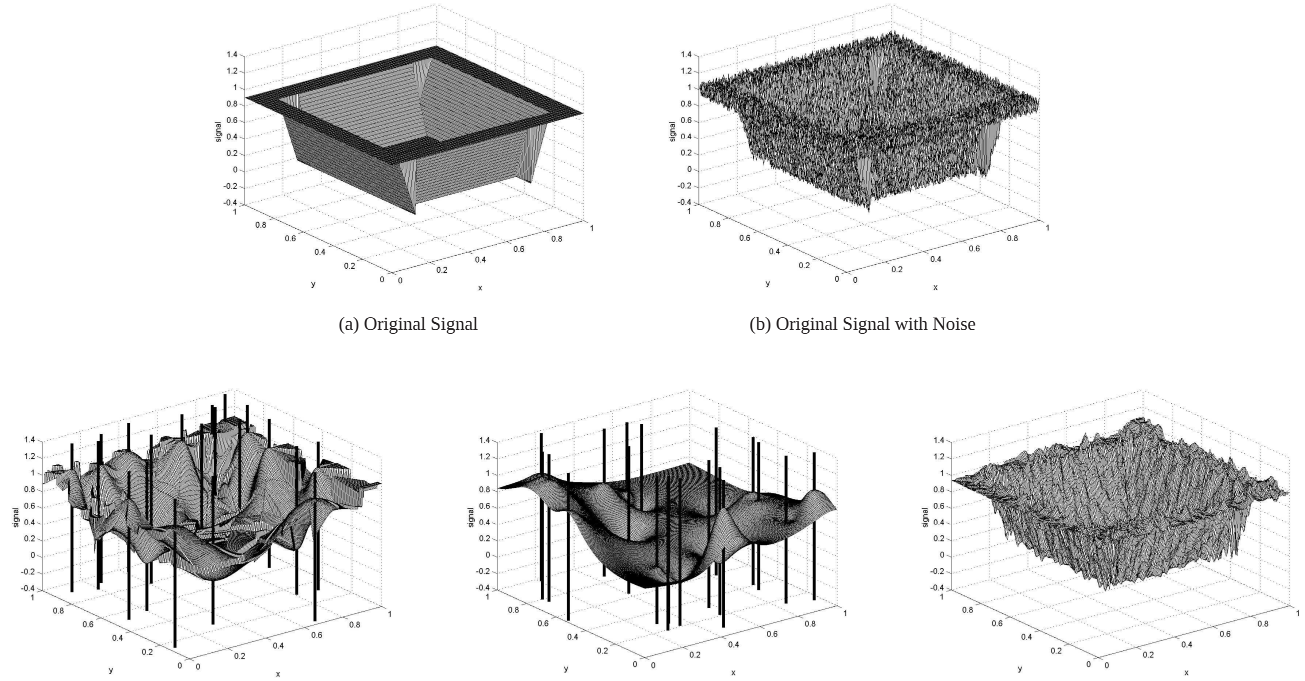
5. Conclusions

In this paper we address a problem of paramount importance: how to compute aggregate quantities in large-scale dense sensor-rich networks. We advocate that mechanisms such as the proposed Physical Dynamic Priority Dominance ((PD)^2) protocol can be used to devise distributed algorithms able to compute simple aggregate quantities such as MIN, MAX and even less obvious ones such as COUNT, with an extremely low time-complexity. With the use of such a mechanism, MIN (or MAX) can be computed with a time-complexity equivalent to the time to transmit a single value. We illustrated a further use of this paradigm by a brief description of our ongoing work to compute approximate interpolations of sensor data.

(PD)^2 based approaches are a significant example where communications and computations are tightly connected with the physical environment. One of the key aspects of computing aggregate quantities with (PD)^2-based approaches is scalability. Our experience with an ongoing work that exploits such mechanisms on hardware platforms has led us to conclude that this research direction is very promising.

This work focused on a single broadcast domain. Our previous work [16] has however shown how to compute MIN and MAX in a network where a single broadcast does not reach all sensor nodes. The main idea is simply to form clusters such that all nodes in a cluster are in a single broadcast domain, perform the aggregated computation in each cluster and then perform convergecast between all cluster heads. The same idea can be used for obtaining an interpolation of sensor readings even in a network where a single broadcast does not reach all sensor nodes.

Network coding [19] is a recently proposed technique for improving throughput in multihop networks. The main idea is to forward a packet which is not identical to an incoming packet; instead the forwarded packet is a function (for example average values) of incoming packets. Our approach can also be combined with network coding. Consider the example where we are interested in finding both the MIN and the MAX of all sensor readings in the entire network and consider a network which is not a single broadcast domain. We can cluster nodes and apply the MIN and MAX algorithm in each broadcast domain. We can now consider the network as a network of only cluster heads (and some extra nodes to ensure that cluster heads are connected). We are now facing a network where each cluster head can be



(a) Original Signal

(b) Original Signal with Noise

(c) Improved Scheme ($k = 20$)

(d) Random Data Points ($k = 20$)

(e) Improved Scheme ($k = 1000$)

Figure 2. Interpolation Example

treated as a source node for MIN and MAX and the destination of these flows are a sink node. Network information coding can be applied on that network

References

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

[2] International Telecommunication Union (ITU). The internet of things. In *ITU Internet Reports 2005*, November 2005.

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

[4] 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.

[5] M. Weiser. The computer for the twenty-first century. *Scientific American*, pages 94–100, September 1991.

[6] 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.

[7] 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.

[8] S. Madden, M. J. Franklin, J.M. 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.

[9] K. S. J. Pister, J. M. Kahn, and B. E. Boser. Smart dust: Wireless networks of millimeter-scale sensor nodes, 1999.

[10] R. Zheng, L. Sha, and W. Feng. MAC layer support for group communication in wireless sensor networks. In *Proceedings of the second Mobile Adhoc and Sensor Systems Conference (MASS'05)*, page 8. IEEE, 2005.

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

[12] 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 (to appear)*, May, 2008. TR available at http://www.hurray.isep.ipp.pt/privfiles/HURRAY_TR_061102.pdf.

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

[14] 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.

[15] 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.

[16] B. Andersson, N. Pereira, and E. Tovar. Exploiting a prioritized MAC protocol to efficiently compute min and max in multihop networks. In *Proceedings of the 5th Workshop on Intelligent Solutions in Embedded Systems (WISES'07)*, Madrid, Spain, June 2007.

[17] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference*, pages 517 – 524, 1968.

[18] B. Andersson, N. Pereira, S. Prabh, and E. Tovar. Obtaining measurements of non-smooth sensor data, 2008. IPP-HURRAY Technical Report HURRAY-TR-080301, available at http://www.hurray.isep.ipp.pt/privfiles/HURRAY_TR_080301.pdf.

[19] R. Ahlswede, Ning Cai, S.-Y.R. Li, and R.W. Yeung. Network information flow. *Information Theory, IEEE Transactions on*, 46(4):1204–1216, Jul 2000.