Distributed Sensing of Fluid Dynamic Phenomena with the XDense Sensor Grid Network

João Loureiro
Raghu R.
Eduardo Tovar

CISTER-TR-150602

2015/08/19

Distributed Sensing of Fluid Dynamic Phenomena with the XDense Sensor Grid Network

João Loureiro, Raghu R., 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: joflo@isep.ipp.pt, raghu@isep.ipp.pt, emt@isep.ipp.pt
http://www.cister.isep.ipp.pt

Abstract

XDense is a proposed wired mesh grid sensor network system tailored for scenarios that benefit from thousands of sensors per square meter. XDense has scalable network topology and protocols, customizable to application specifics, that enables complex feature extraction in realtime from observed phenomena by exploiting communication and distributed processing capabilities of such network topologies.XDense has been designed with closed-loop CPS applications like active flow control of aircraft wing surfaces in mind. It uses a plug-n-play architecture that allows dimensioning of application specific networks. In this paper, we evaluate the performance of XDense in a fluid dynamic application scenario. With experiments on feature detection and realtime scenarios, we demonstrate the potential of the architecture and discuss practical implementation issues.

Distributed Sensing of Fluid Dynamic Phenomena with the XDense Sensor Grid Network

João Loureiro, Raghuraman Rangarajan, Eduardo Tovar
CISTER/INESC-TEC, ISEP, Polytechnic Institute of Porto, Porto, Portugal
Email: {joflo, raghu, emt}@isep.ipp.pt

***Abstract*—XDense is a proposed wired mesh grid sensor network system tailored for scenarios that benefit from thousands of sensors per square meter. XDense has scalable network topology and protocols, customizable to application specifics, that enables complex feature extraction in realtime from observed phenomena by exploiting communication and distributed processing capabilities of such network topologies. XDense has been designed with closed-loop CPS applications like active flow control of aircraft wing surfaces in mind. It uses a plug-n-play architecture that allows dimensioning of application specific networks. In this paper, we evaluate the performance of XDense in a fluid dynamic application scenario. With experiments on feature detection and realtime scenarios, we demonstrate the potential of the architecture and discuss practical implementation issues.**

I. INTRODUCTION

The advent of microelectromechanical systems (MEMS) has enabled new applications to be developed that rely on dense deployments of sensors , with granularity as small as few micrometers of sensor inter space and sampling rates up to kilohertz. For deployments as dense as thousands of nodes in a few square meters area, sensor network technology faces scalability issues in many key aspects as such as cost, communication time, interconnectivity, processing time, power, and reliability [1]. Processing all this data for feature extraction becomes costly and extraction is difficult to achieve in real time, hence prohibiting its use in real time applications like closed-loop actuation.

There are many CPS applications of this phenomena that require real-time data for actuation scenarios. For example, flow control on aircrafts [2] and under-water vehicles [3]. There are many other application examples for such deployments ranging from artificial skins for robotics [4], to biomedical devices such as implantable prosthesis [5]. In this paper we focus on fluid dynamic phenomena, which have tight spatial and temporal sensing requirements [2].

XDense is proposed as a network architecture tailored to address the challenges of such extremely dense sensor deployments. It differs in some ways from traditional wireless sensor network approaches: We use point-to-point (P2P) links, which are not susceptible to concurrency or noise issues like shared buses or wireless topologies are, and moreover allows higher communication rates; We consider a far more denser deployment scenario than traditional SNs (thousands of nodes per square meter); Nodes may share power supply and the impact of communication on power is negligible compared to battery powered radio links;

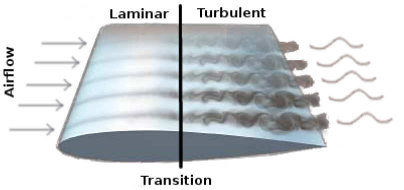


Fig. 1: Example fluid dynamic application scenario: air flow over a wing surface exhibiting transition from laminar to turbulent flow.

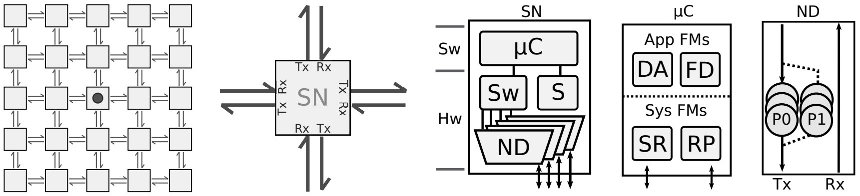
XDense is inspired by Network-on-Chip (NoC) architecture, and is composed of regular structures that form mesh grids. Also has similarities in routing schemes, timing properties and distributed computing capabilities [6]. On the other hand, the network is not on a single chip, but built on a larger surface that is physically attached, specific to each application scenario, and the node count is greater than that for NoC applications; Also, the input data is generated at each node by its sensor (which imposes different restrictions and opportunities).

Our contribution: XDense enables efficient data extraction of observed phenomena without the need of collecting the data from each individual node centrally. Instead, it takes advantage of its mesh grid topology to allow the user to program the node's data extraction algorithms with respect to the application's objectives.

We believe XDense differs from available SN architectures since it was conceptualized and designed specifically to maximize opportunities on distributed processing for demanding applications, keeping scalability and low system complexity in mind. In a sense, XDense tries to build a SN that behaves as if it is a distributed many-core computation platform, but programmed targeting specific sensing applications. Note that, the aim in this paper is to show off the capabilities of the architecture itself and moving towards the realisation of a testbed, and not demonstrate performance analysis of the protocols (hence the protocols have been kept simple).

We first conceptualized XDense with some preliminary simulation results in [7]. Lately in [8] we presented a hardware prototype using COTS, and the results of some basic experiments. Initial simulation results were also presented in [9]. In this paper we use simulation to evaluate the benefits of using simple feature detection and extraction algorithm in our system, for reducing transmissions for faster response times.

Roadmap: In Section II we first discuss dense sensor deployment scenarios for fluid dynamic applications. In Section III, we review relevant related work. In Section IV, we highlight the significant aspects of XDense architecture



(a) (b) (c) (d) (e)

Fig. 2: Overview of XDense architecture (a) A 5×5 network with one sink in the center; (b) Node pinout: two channels per port for transmitting and receiving data; (c) Node's model architecture: a software layer, which is the application, and the hardware layer, which includes the switch (Sw), the net-device (ND) and the sensor (S); (d) Function Modules (FMs) utilized in our evaluation include System FMs for essential functions and Application FMs for application specific algorithms.

and protocols. In Section V, we evaluate the performance of XDense. Section VI and VII comment on dimensioning issues and outlines our conclusions and planned steps forward.

II. DENSE SENSOR DEPLOYMENTS FOR FLUID DYNAMIC APPLICATIONS

An important objective in fluid mechanics is to characterize airflows and identify the presence of turbulence, since it governs important properties of these flows such as their spreading, mixing, and the way it interact with its surrounding [10]. Turbulence can be highly undesirable on aircrafts considering that it may increases drag and noise, and consequently fuel expended [11], and a common goal is its minimization. Figure 1 shows an airflow phenomena over a wing surface and illustrates the transition from laminar to turbulent flow. This transition separates laminar flow (which has a more homogeneous speed profile distribution) from turbulent flows (which is composed of coherent structures) [12]. An increase in the turbulent region consequently increases drag on the surface of the wing.

Some techniques allow study of a flow's properties by extracting profile data such as speed and temperature. For that, large deployments of sensors may require sensors' interspace to be smaller than that of the spatial granularity of the observed phenomena (for example, of 100 μm or less) and have high sampling rates (in excess of 10 kHz) [13]. Kasagi and others [2] surveyed micro flow sensors that attempt to meet such requirements, for example with sensor arrays. Validations are usually done offline by analyzing logged data of the sensors individually connected to channels of an multiplexed analogue-to-digital converter (ADC) [14]. Moreover, there are applications that demand real-time data to perform closed loop actuation for active flow control (AFC), which leads to even tighter requirements. Cattafesta and others survey actuators for such scenarios and opportunities for AFC [15].

Based on the above discussion, we state the following requirements of such a system: (a) Efficient data extraction: The network infrastructure should allow efficient extraction of complex information about the phenomena without the need of centralization and processing of data. (b) Scalable infrastructure: An increase in sensor count should have minimal impact on the complexity of architecture (spatial and temporal). (c) Real time behaviour: The network should be able to respond in a timely manner, such that actions can be taken based on the extracted data.

III. RELATED WORK

For its operation, XDense relies on subsets of operating principles of distinct systems. For example,

our network architecture resembles Array processors [16] that are widely found in literature and inside industry. This kind of architecture brings advantages due to its modularity and scalability (based on regular structures). For example Geometric-arithmetic Parallel Processor (GAPP) is a parallel processor with over 10,000 processing elements [17]. It consisted of very simple processing elements (one bit sum only) to perform matrix operations. Although the interconnection arrangement is similar to our work, the processing elements are passive, and are not a network of active programmable nodes as conceived for our design.

Grid networks of more complex processors have now become commercially available [18] and allow complex distributed data processing algorithms to explore the many-core computational potentials in a transparent way. This Systems-on-chip (SoC) have proven benefits. Further, they operate analogous to our sensor network as a whole, except that its projected to be confined into a single chip, what brings different limitations and opportunities.

On the other hand, wireless sensor networks (WSN) are suitable for large scale deployments, and effort have gone towards minimizing communication by distributively processing data. For example, in [19], the authors explore distributed data compression. Other works employ techniques that, similar to the works listed, rely on local communication between neighbors to detect contours [20]. But wireless sensor nodes were not designed for the scenarios we are interested in, and they exceed in complexity, power constraints, size, latency and overhead, and therefore does not meet the application temporal requirements.

Closer to our architecture, a multi modal sensor network was proposed in [21]. It is a sensor network with an embedded processor dedicated to each sensor node, which communicate with its surroundings using an infrared transceiver. But, due to link contentions and collisions, in other words, cost of communication, their research leans more towards WSNs, which are not suitable for the applications we are focused on.

IV. XDENSE ARCHITECTURE

The selection of a good topology is a job of fitting network requirements to available technology. A trade-off between cost

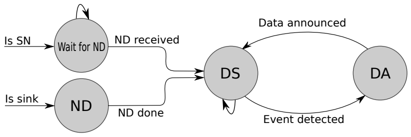


Fig. 3: State diagram for an XDense node.

and performance should always be considered when specifying many aspects. Considering this, and the requisites from Section II, we now briefly describe the XDense architecture and protocols in this section. More detailed explanation can be found in previous publications [9].

A. Architecture Details

Our architecture consists of a 2D mesh network of sensor nodes and sinks, with point-to-point wired connections with up to four neighboring nodes, physically located in four directions (see Figure 2(b)). Figure 2(a) shows an example scenario of a 5×5 network (24 nodes and one sink) with the sink located in the center.

Figure 2(c) shows the main functional blocks of a sensor node (SN). Each node can be seen as a system on chip (SoC), with dedicated hardware peripherals and a CPU (μC). The Switch (*Sw*) and the Net-Devices (*ND*) are responsible for communication on the network, and the Sensor (*S*) is the sensing element of each node. Each of these components are detailed in the following paragraphs.

The network is homogeneous in its components, apart from the sinks. The sink may have an external link, for example, to a wireless connection to a supervisory system, or to a local actuator for closed loop distributed actuation. More than one sink may be present in the network, with each one responsible of collecting data from its address region, which is a configurable system parameter.

At the bottom layer in the node's architecture (Figure 2(e)), Net-Devices (*ND*) connect two distinct nodes. Each node contains four *ND*s that connects them to their four immediate neighbors in the grid. Each *ND* consists of a full-duplex serial port , with two output queues (this is discussed later and are referred as *p0* and *p1*). The switch (*Sw*) is the interface between *ND*s and the application layer. It connects n *ND*s to the application and allows individual or parallel access to any *ND*. It is able to store and forward packets among *ND*s without interference of the application layer.

The μC can be seen as a set of building blocks for applications (Figure 2(d)). These building blocks or function modules (*FM*) implement different functionalities for the network's operation. There can be two types of *FM*s: System *FM*s that setup essential functionalities, and Application *FM*s that implement algorithms specific to an application. One or more sensors interact with the physical world and are connected to the μC through it's analogue-to-digital interface. The μC should be able to interface with one or more sensors of different natures, consonant to the application's monitoring goal.

B. Operational States

The distributed protocol consists of three operating states (Figure 3). First the sink floods an *Network Discovery (ND)* packet carrying the system's settings about sink location (or packet origin), baudrate, sampling rate, and the functional modules to be used. After receiving at least one *ND* packet, nodes switch to the *Data Sharing (DS)* state. In the *DS* state, SNs continuously sense the environment at the configured sampling rate, and communicate the sensed values with their neighborhood (defined by parameter n_{hops}). That is, all SNs initially send their values in all four directions, which is then stored and forwarded by the immediate neighboring nodes up to the n_{hops} distant neighbor. Packets exchanged during this state are queued in the lower priority queue (*p0*) of the Net-device. The naive case is $n_{hops} = 0$, which is each node sending data directly to the sink without communicating with its local neighborhood. Computations are done on collected data in order to detect it's characteristics. If the node computes data of interest it switches to *Data Announcement (DA)* state.

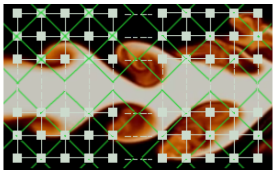


Fig. 4: XDense network superimposed on the CFD dataset snapshot from [22], showing clustering for n_{hops} = 1.

On switching to the *DA* state, a node forwards the packet with information detected towards the closest sink. In turn, the sink receives data from different origins, allowing it to reconstruct the observed phenomena with increasing accuracy and coverage (after each *DA* reception). Packets exchanged during this state are queued in the higher priority queue (*p1*).

V. EVALUATING XDENSE WITH CFD DATA AS INPUT

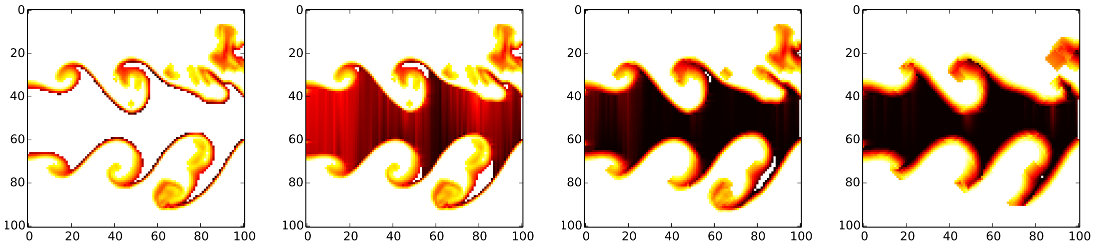
In order to construct the XDense sensor network we analyze the performance of the XDense architecture with a simulation study. We perform two experiments to evaluate XDense and study its distributed processing capabilities.

In some application scenarios, detecting just the features of the flow is enough to reach a decision (for example, edges of a flow in active flow control). As we will show with in-network feature detection, and the trade-off between different network configurations. This experiment is static in nature and analyze only one snapshot of an input phenomena. In the second experiment, we look at the above application function with temporal data as input.

A. Input phenomena and performance metrics

We first need to simulate the sensing of a real world phenomena using our network. We achieve this by performing a simulation that integrates the network model with computational fluid dynamics (CFD) data as its input. That is, we "feed" each sensor of our network with spatial and temporal data extracted from a reliable representation of a real phenomena using CFD.

For the representative input phenomena, we utilize the results of a CFD scenario proposed to study a flow that simulates a planar jet emitting from a nozzle (into a tank filled with same fluid) [22]. A single time instant snapshot showing



(a) $n_{hops} = 1$ (b) $n_{hops} = 1$ (c) $n_{hops} = 2$ (d) $n_{hops} = 4$

Fig. 5: Feature detection: (a) Extracted boundary data for $n_{hops} = 1$, (b) reconstruction of boundary data for the same scenario. (c) and (d) shows reconstructed data for $n_{hops} = 2$ and $n_{hops} = 4$ for comparison.

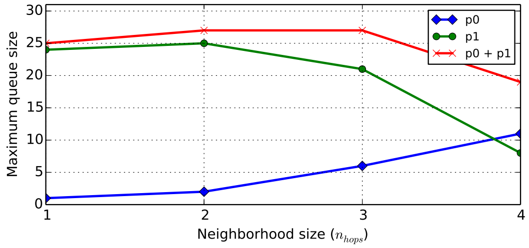


Fig. 6: Maximum queue size for $n_{hops} = 1$ to 4.

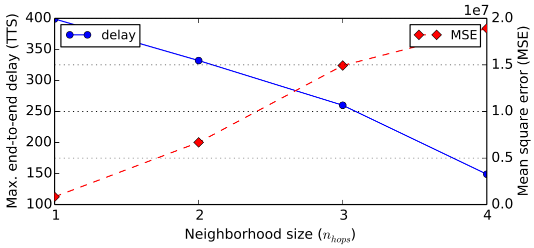


Fig. 7: Trade-off between mean square error and maximum end-to-end delay for different values of n_{hops}.

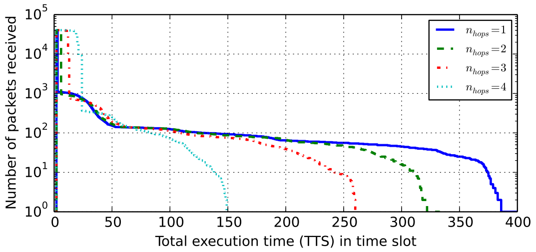


Fig. 8: Number of receptions over time for different values of n_{hops}.

the transport of a flow scalar superimposed on the sensor network is shown in Figure 4. This scenario, and the wing scenario from Figure 1, present similar mixing phenomena of laminar and turbulent flows. By using this CFD data as input, we demonstrate the possibility of detecting application specific features, such as transition regions, as of common interest in avionics applications.

We used NS-3 for the implementation and extended it by developing a module for NoC-like grid networks. We also implemented the abstraction layers of our communication protocols and function modules (see Figure 2(c) and 2(d)).

The SN deployment inter-space depends on the minimum size of the observed phenomena, and therefore has to be smaller then the minimum turbulent structure size [13]. The number of sensor nodes are chosen based on the area to be covered and we use a $101 \times 101 - 1 = 10200$ SN with one sink in the center. The baudrate and sensor's sampling rate are also based on the application requirements and for this reason we normalize our temporal results to time slots. We use two parameters for this. A transmission time slot (TTS) is the time required to transmit one packet and a sampling time slot (STS) is the period between two consecutive samples. Prioritized queues (*p0* and *p1*) allow *DA* packets to have priority over *DS* packets for better response times.

We analyze the performance of XDense in terms of end-to-end delay, load on the network, queue size and quality of acquired data, what helps on evaluating dimensioning issues.

B. Feature Detection

In this experiment we use a distributed feature detection algorithm to detect the transition layer of flows. We use an image processing based algorithm [23] with this propose. This is, an *FM* with a variation of the Sobel [24] operator for edge detection, which is widely utilized in the image processing domain. In brief, each node performs a 2-D spatial gradient measurement on an image and emphasizes regions of high spatial frequency that correspond to edges.

A node applies the Sobel operator to its data and checks if it is above a threshold which defines the sharpness of the edge that triggers edge detection (as defined by the application). If an edge is detected, the node uses a *FM* that performs linear regression with the neighborhood data. This information is transmitted to the sink, which is then able to aggregate the overall scenario with the information received from distinct regions of the observed phenomena. For reconstructing the picture of the phenomena, based on the aggregated data, the sink performs linear interpolation around the center region of the jet, between the upper and lower limits that define the body of the flow.

Figure 5 shows the results of feature detection for different values of n_{hops}. Figure 6 show that the greater the n_{hops} is, the smaller is the sum of both queues *p1* + *p2*. And from Figure 7 the greater the n_{hops} is, the end-to-end delay is minimum, but with a cost of resolution, with maximum mean square error (MSE). There is a trade-off due to loss of high frequency shapes on the phenomena, like the vortices's indentations. End-to-end delay is shown in Figure 8. For $n_{hops} = 1$, the delay drops by a factor of 10 when compared to naive scenario,

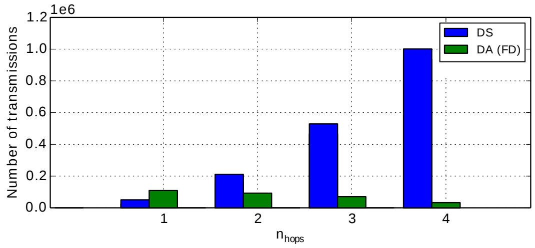


Fig. 9: Total number of packets transmitted in the network with $n_{hops} = 1$ to 4.

which takes 10200 transmission time slots (TTS) for the sink to receive data from all the nodes. Figure 9 shows the balance between the total number of packets exchanged during *DS* and *DA* states. By increasing local communication, we decrease transmissions to the sink, and end-to-end delay.

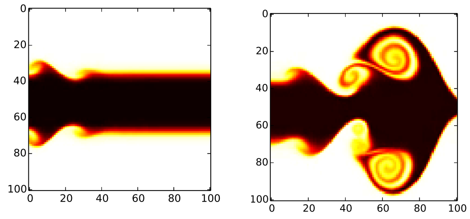
We can compare our results with Pressure Belt, a master-slave, shared bus based network proposed in [25], for monitoring pressure over an aircraft wing. [1] The authors calculate the time required to read all the nodes simply as: *number of nodes*×*packet duration*.

Normalizing Pressure Belt results, and considering one master for each quarter of all nodes, Pressure Belt would have the same performance of the naive operational scenario mentioned above, in which all nodes switch to *DA* state to announce its own raw reading of its sensors. But, due to physical and electrical limitations, and due to the maximum address space, their solution may not be suitable for this density of nodes. Moreover, their solution was designed for offline processing, acting as a simple data logger, making it inapplicable for real-time applications.

C. Realtime sensing

As stated before, the goal of XDense is to sense data in real time and enable actuation based on that information. We now extend the previous analysis, which had static input data, to temporal data.

Figure 10 shows the input phenomena evolution for a period of 120 samples. Within this period, a turbulence with vortices emerges in the flow phenomena and our aim in this experiment is to see the reaction of the network to this data in real time. Figure 11(a) presents the network activity with respect to time. The network continuously monitors the evolution of the flow during this time period. There is an increase in network activity as the flow becomes more turbulent, but the steepness is determined by the algorithm and by n_{hops}. A common pattern of peaks can be identified. This is caused by nodes reacting similarly to abrupt variations on the input data, as the appearance of a new vortices's in the observed area. Moreover, for a stable system, we have to make sure that one entire cycle of *DS* followed by *DA*s is contained in between two sensor samples.



(a) $t_{\text{initial}} = 0$ (b) $t_{\text{final}} = 120$

Fig. 10: Evolution of input data from (a) t=0 to (b) t=120 sampling time slots (STS).

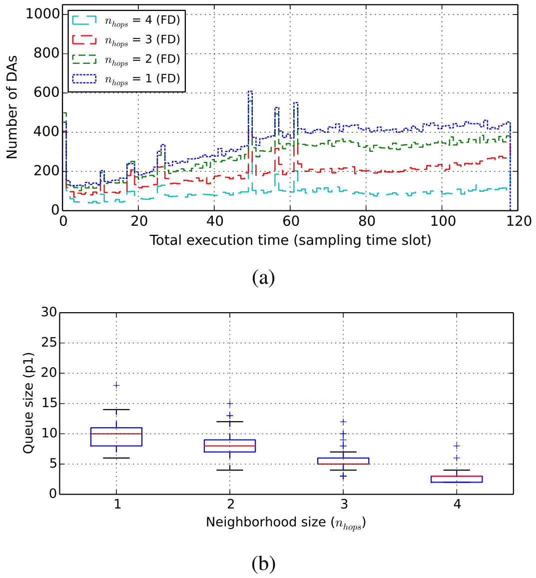


Fig. 11: (a) Network activity, shown as number of *DA*s per sampling period, over 120 sampling periods, for $n_{hops} = 1$ to 4. (b) *p1* queue sizes distribution for $n_{hops} = 1$ to 4

Figure 11(b) shows the distributions of the maximum *p1* queue size per sampling period (from t=0 to t=120). Queue mean size is inversely proportional to the value of n_{hops}, with more contained variations and bounded behavior with the growth of n_{hops}.

VI. OBSERVATIONS ON DIMENSIONING XDENSE

Dimensioning is an important issue in designing XDense and this depends on the application scenario. Issues of network size, communication rate and node capabilities, functional modules ideal for a particular application scenario, are all a part of dimensioning.

Choosing the right *FM*s is the job of the design engineer and is highly application dependant. Designers should be able to design any distributed processing algorithms, with the premise that local communication costs are much lower than cost of communicating with the sink. Generic applications can be developed, for example, to compress data of any kind using standard lossy/lossless image compression algorithms like MPEG, to transmit compressed streams of the sampled data.

---

[1] Pressure Belt [25] consists of a strip mounted crosswise on the wing of an aircraft, connected in one extremity to a coordinator, and has an embedded data-logger situated inside the airplane. It runs over two parallel full-duplex RS485 shared links, compatible with the IEEE 1451.2 Standard for a Smart Transducer Interface for Sensors and Actuators. Up to 255 nodes can communicate at 5 Mbps with packets of 48 bits each. By using a clock synchronization scheme, time division multiple access (TDMA) is used to communicate with the nodes.

Specific application-oriented issues can also be thought of, for bringing even more performance potentials to this kind of architecture. Edge detection algorithms allows filtering high frequency components in space. In fluid dynamics, a flow's parameters, like the Reynolds number, could be distributively inferred based on analysis of the vortex characteristics [26].

VII. CONCLUSION

XDense-like dense sensor networks are going to be increasingly used in application scenarios which require realtime data. Combined with novel feature detection techniques, systems can then recreate the phenomena present in the application scenarios, and use this information for realtime actuations. XDense provides a clean framework for addressing such requirements.

We evaluated the XDense architecture and protocols with metrics on accuracy, timeliness and network usage and showed the tradeoffs that have to be made among the system parameters. It is important to further understand the underlying issues that affect the performance of XDense. Dimensioning XDense concerns issues such as node density and neighborhood size for information sharing. Also, the implemented function modules affect performance of the system and depend closely on the application scenario.

As stated initially, the protocols were not the focus of the paper and the aim was to demonstrate the suitability of the architecture for realising a practical testbed. The chosen application scenario will dictate the actual protocols used and will be the focus of future work.

It is important to note that the simplicity of the architecture allows for practical construction of such networks using currently available technology. The simulations in this work establishes the competence of XDense and our next step is developing a prototype for experimental evaluation. A low-cost solution was developed with COTS microcontrollers, with preliminary results were presented in [8].

ACKNOWLEDGMENTS

This work was partially supported by National Funds through FCT/MEC (Portuguese Foundation for Science and Technology) and when applicable, co-financed by ERDF (European Regional Development Fund) under the PT2020 Partnership, within project UID/CEC/04234/2013 (CISTER Research Centre); also by FCT/MEC and the EU ARTEMIS JU within project ARTEMIS/0004/2013 - JU grant nr. 621353 (DEWI, www.dewi-project.eu); by the government of Brazil through CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico.

REFERENCES

[1] D. Ganesan, A. Cerpa, W. Ye, Y. Yu, J. Zhao, and D. Estrin, "Networking issues in wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 799–814, 2004.

[2] N. Kasagi, Y. Suzuki, and K. Fukagata, "Microelectromechanical systems-based feedback control of turbulence for skin friction reduction," *Annual review of fluid mechanics*, vol. 41, pp. 231–251, 2009.

[3] S. L. Ceccio, "Friction drag reduction of external flows with bubble and gas injection," *Annual Review of Fluid Mechanics*, vol. 42, pp. 183–203, 2010.

[4] K. Takei, T. Takahashi, J. C. Ho, H. Ko, A. G. Gillies, P. W. Leu, R. S. Fearing, and A. Javey, "Nanowire active-matrix circuitry for low-voltage macroscale artificial skin," *Nature materials*, vol. 9, no. 10, pp. 821–826, 2010.

[5] J. Ohta, T. Tokuda, K. Sasagawa, and T. Noda, "Implantable cmos biomedical devices," *Sensors (Basel, Switzerland)*, vol. 9, no. 11, p. 9073, 2009.

[6] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, and A. Hemani, "A network on chip architecture and design methodology," in *VLSI, 2002. Proceedings. IEEE Computer Society Annual Symposium on*. IEEE, 2002, pp. 105–112.

[7] J. Loureiro, V. Gupta, N. Pereira, E. Tovar, and R. Rangarajan, "Xdense: A sensor network for extreme dense sensing," *Proceedings of the Work-In-Progress Session at the 2013 IEEE Real-Time Systems Symposium*, pp. 19–20, 2013.

[8] J. Loureiro, R. Rangarajan, and E. Tovar, "Demo abstract: Towards the development of xdense, a sensor network for dense sensing," *12th European Conference on Wireless Sensor Networks (EWSN)*, p. 23. [Online]. Available: www.cister.isep.ipp.pt/ewsn2015/EWSN15PosterDemoProc_WEB.pdf#page=24

[9] ——, "Xdense: A dense grid sensor network for distributed feature extraction," *33o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2015) - Workshop de Comunicação em Sistemas Embarcados Críticos (WoCCES)*, 2015.

[10] S. Corrsin and A. L. Kistler, "Free-stream boundaries of turbulent flows," DTIC Document, Tech. Rep., 1955.

[11] W. K. Blake, *Mechanics of Flow-Induced Sound and Vibration V2: Complex Flow-Structure Interactions*. Elsevier, 2012, vol. 2.

[12] S. K. Robinson, "Coherent motions in the turbulent boundary layer," *Annual Review of Fluid Mechanics*, vol. 23, no. 1, pp. 601–639, 1991.

[13] U. Buder, R. Petz, M. Kittel, W. Nitsche, and E. Obermeier, "Aeromems polyimide based wall double hot-wire sensors for flow separation detection," *Sensors and Actuators A: Physical*, vol. 142, no. 1, pp. 130–137, 2008.

[14] A. Berns and E. Obermeier, "Aeromems sensor arrays for time resolved wall pressure and wall shear stress measurements," in *Imaging Measurement Methods for Flow Analysis*. Springer, 2009, pp. 227–236.

[15] L. N. Cattafesta III and M. Sheplak, "Actuators for active flow control," *Annual Review of Fluid Mechanics*, vol. 43, pp. 247–272, 2011.

[16] S. Y. Kung, "Vlsi array processors," *Englewood Cliffs, NJ, Prentice Hall, 1988, 685 p. Research supported by the Semiconductor Research Corp., SDIO, NSF, and US Navy.*, vol. 1, 1988.

[17] E. Cloud, "The geometric arithmetic parallel processor," in *Frontiers of Massively Parallel Computation, 1988. Proceedings., 2nd Symposium on the Frontiers of*, Oct 1988, pp. 373–381.

[18] *Epiphany 64-core Microprocessor*, Adapteva, 2014, rev. 2.

[19] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," in *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 2009, pp. 145–156.

[20] M. Li and Y. Liu, "Iso-map: Energy-efficient contour mapping in wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 5, p. 699710, May 2010.

[21] J. Lifton, D. Seetharam, M. Broxton, and J. Paradiso, "Pushpin computing system overview: A platform for distributed, embedded, ubiquitous sensor networks," in *Pervasive Computing*. Springer, 2002, pp. 139–151.

[22] C. B. da Silva and J. C. Pereira, "Invariants of the velocity-gradient, rate-of-strain, and rate-of-rotation tensors across the turbulent/nonturbulent interface in jets," *Physics of Fluids (1994-present)*, vol. 20, no. 5, p. 055101, 2008.

[23] J. Westerweel, C. Fukushima, J. M. Pedersen, and J. Hunt, "Momentum and scalar transport at the turbulent/non-turbulent interface of a jet," *Journal of Fluid Mechanics*, vol. 631, pp. 199–230, 2009.

[24] O. Vincent and O. Folorunso, "A descriptive algorithm for sobel image edge detection," in *Proceedings of Informing Science & IT Education Conference (InSITE)*, 2009, pp. 97–107.

[25] F. Gen-Kuong and B. Swanson, "Network sensor systems- the pressure belt application," in *19 th Aerospace Testing Seminar, Manhattan Beach, CA*, 2000, pp. 379–390.

[26] K. Iwamoto, Y. Suzuki, and N. Kasagi, "Reynolds number effect on wall turbulence: toward effective feedback control," *International journal of heat and fluid flow*, vol. 23, no. 5, pp. 678–689, 2002.