Data Validity and Dependable Perception in Networked Sensor-Based Systems

Luís Marques
lmarques@lasige.di.fc.ul.pt
FC/UL

António Casimiro
casim@di.fc.ul.pt
FC/UL

***Abstract*—Although the technology and applications of wireless sensor networks have greatly increased over the last years, ensuring a dependable real-time operation despite faults and temporal uncertainties is still an on-going research topic. The problems are particularly significant when considering that future applications will interact with their environment not only for supervision or monitoring, but also to directly control physical (real-time) entities, sometimes with safety-critical requirements.**

We believe that reasoning in terms of data validity might be a good way to approach the problem. The ability to know if sensor data flowing in the system is valid – data validity awareness –, is a first step to achieve a dependable operation. But more than that, it should be possible to ensure, given requirements for data validity throughout the operation, a dependable perception of the environment.

In this paper we essentially discuss the problem, analyzing some of the issues that need to be addressed to achieve these goals. Particularly, we introduce fundamental concepts and relevant definitions, we elaborate on the main impediments to achieve data validity awareness and describe relevant means to deal with these impediments. Finally, we address the issue of ensuring a dependable perception and present some research ideas in this direction.

I. INTRODUCTION

Networked sensor-based systems, such as those referred to as "wireless sensor networks", "cyber-physical systems" and "Internet of things", refer to systems where information technology pervades and closely interacts with the real world environment. The real-time nature of information processed in these systems creates difficult problems, namely how to ensure that the system possesses temporally and logically valid data to work with. One intuitive way of dealing with this problem is to provide the system with mechanisms to ascertain the validity of data and thereby be able to reject invalid data – that is, to provide *data validity awareness*. Nevertheless, note that this only prevents bad data from being used – it does not guarantee that good data will be eventually available.

The concept of data validity is usually not relevant in real-time systems. Instead, these systems are designed to secure specific temporal constraints, which are derived, among other, with the objective of ensuring data validity at all times. Thus, data is always valid by definition and by construction. These systems are engineered using several techniques for synchronous designs, and they may become dependable sometimes at the cost of requiring many resources like when using high sampling rates, redundant components or redundant operations.

Unfortunately, in many types of networked sensor-based systems it is often not possible to follow these approaches. On the one hand, the very fundamental design of the system is permeated with uncertainty, in particular caused by the employment of wireless ad hoc networking, which is much more susceptible to interferences than fixed networks. This uncertainty, often further complicated by issues such as node mobility and topology changes, is the enemy of a predictable system. Therefore, it is often impossible to provide guarantees about the temporal validity of data. Furthermore, the interaction with a myriad of sensors of all kinds, in an increasingly open and ad hoc environment, creates more opportunities for faults in sensors to introduce erroneous data in the system and thus affect the (logical) validity of data. On the other hand, the traditional techniques used in real-time systems to ensure predictability and reliability are often too costly or inadequate. High sampling rates, frequent communication and redundant channels are detrimental to many battery limited sensor-based networks, such as wireless sensor networks, while temporal redundancy may be penalizing to the validity of data.

We believe that a different look at how to build these systems is necessary and that reasoning about the validity of data is a potentially good approach. Rather than assuming bounds for fundamental variables (like processing times, communication delays or faults), thus having a somehow fixed view of how an environment will look like, it seems more reasonable to encompass uncertainty by endowing the system with means to ascertain the validity of data at any given moment or, better than that, providing means to raise awareness of the achievable data validity in any given environment.

Although there is past work in this direction, which we intend to explore, there are still many open questions with respect to the objective of achieving dependable real-time operation in networked sensor-based systems. In this position paper we provide an outlook of several issues that are on the way to develop solutions for the problem. Our main contributions are: a) a systematization of the main impediments to data validity awareness, in particular, and perception dependability in general; b) a description of some notable approaches and mechanisms to provide applications with awareness about data validity; c) a discussion of open issues with respect to achieving perception dependability and the provision of research ideas and solutions to address them.

Before we provide some important concepts and definitions, and before we delve into the main issues of the paper, in the next section we overview some related work, in which real-time or predictability concerns despite uncertainties affecting the operation have also been considered.

II. Related Work

The challenges in predictability and efficiency in network sensor-based systems start at the lowest levels. Low power systems, such as wireless sensor networks, have limited communication coverage and have to be densely deployed, which leads to the need of efficient MAC layer protocols. At this level inefficiencies are mostly the result of packet collisions, overhearing packets destined to other nodes, control packet overhead and idle listening [1]. Different protocols have been proposed to address these challenges (see [1] for an overview), but no protocol has been accepted as a standard, as no single one has been found best for all types of applications. The available protocols have different efficiencies, latencies and predictability, with different resilience to collisions and interference. Limited research has been done regarding integrating MAC layer protocols with network layer protocol to optimize medium access control based on successful and unsuccessful packet routing and, to our knowledge, no research has looked into optimizing the MAC layer depending on the predictability, efficiency and latency requested at the application level.

Going into the other network layers, several protocols exist which try to prioritize real-time traffic, with different approaches and specific aims.

The EDDD protocol extends [2] the concept of directed diffusion, which uses the application context in data dissemination, with support for routing priorities, resulting in better energy conservation and packet delays.

[3] includes an integrated MAC and network cross-layer protocol for efficient, timely transmission of packets, with dynamic adaptation to application requirements, providing an average reliability using probabilistic parameters.

The RAP architecture [4] tries to provide real-time communication in large-scale wireless sensor networks by implementing a new packet scheduling policy that accounts for both data deadlines and transmission distances, reducing deadline misses for packets far-away from their destination.

The $(\mathrm{RT})^2$ [5] protocol tries to address the real-time and reliability issues occurring in networks where both sensing and actuation occur (sensor and actor networks). Most significantly, it considers the assumption that several sensor readings are necessary for reliability and tries to adapt transmission rates to obtain the necessary number of readings.

[6] improves timing guarantees in highly dynamic sensor networks, such as mobile robotic sensor networks. It argues that the root cause of routing failures in these networks is the delay between the creation of the routes and their use, and proposes delaying route calculation to the latest possible time – when packets are transmitted. Simulations of this lazy-binding routing show a 10 fold improvement of packet delivery in highly mobile sensor network.

All of these protocols improve performance in some specific set of conditions. But, even though some are billed as providing real-time quality, they are either best-effort or they adapt in such a way that doesn't guarantee application-level requirements.

One area where the conflict between timing issues and energy conservation has been particularly onerous is that of multimedia streaming in wireless sensor networks. [7] presents a survey of this field. It investigates the requirements of multimedia streams (high throughput, low end-to-end delay, low jitter, high computational power) and the various solutions available at the application, network and MAC layers. In short, it concludes that existing MAC and network layer protocols present a few different mechanisms that try to improve performance but are still very limited in achieving application-level dependability objectives. As such, the authors suggest cross-layer solutions, which try to map application requirements at the various communication levels.

At the middleware level there is some relevant work for timely and dependable network sensor-based systems. Two important features are support for QoS and support for adaptation. [8] presents a survey of middleware for wireless sensor networks, which includes an analysis of QoS support. [9] offers a survey on adaptable middleware for wireless sensor networks.

QoS in WSNs is identified as still being an open research issue, and that future work should focus on *collective* QoS metrics. That is, not in terms of how data is gathered from individual sensors but from groups of sensors whose information is correlated, due to the dense WSN deployments. This is compatible with the possibility of using data validity as a measure of QoS, since it incorporates the effects of all factors affecting the environment perception. The analysis of [9] also encompasses QoS and it considers the MiLAN middleware as the only one with significant QoS capabilities. Nevertheless, this middleware is oriented towards personal medical monitoring and its QoS mechanisms only tackle the issue of sensor reliability.

Given this brief survey of related work, a general observation is that the typical trend is to find solutions that improve performance and trade-offs, but no solutions are specifically directed to support a dependable operation, for instance by providing awareness of the operational conditions or driving adaptation in order to satisfy a given application bound or requirement (i.e., achieving dependable adaptation).

III. Fundamental Concepts and Definitions

In this paper we are considering a system model composed by sensor nodes, which acquire and propagate sensed environment data; a wireless network, which links the sensor nodes; and one or a few sink nodes, which receive sensor data. Applications execute in sink nodes, and their correctness criteria depend on the validity of sensed data. These applications can be of monitoring or control type and are data centric, in that they are highly dependent on the sensed data. For example, to control robots in industrial environments, which may even need to cooperate, they can behave as a sink of a sensor network which feeds them the positions and velocities of moving objects, but the robots must be aware of the validities of that data for safe operation.

We consider the employment of middleware services in sink nodes, and possibly also on sensor nodes. The services will provide support and facilitate the programming of these data-centric applications, by dealing with mechanisms such as timely propagation of data (at sensor nodes) and assisting in dependable application adaptation (at sink nodes).

One of our main concerns is, therefore, to understand the several issues related to ensuring the validity of sensed data.

Networked sensor-based applications typically have to acquire the state of physical entities (temperature, speed, distance, light intensity, etc), whose values are continuously evolving with time. In the literature, these have been called time-value entities or *Real-Time entities* (RTe) [10]. A RTe is sampled in the system periphery and the value, which is a representation of the RTe at a certain time instant, is kept in a *Real-Time representative* (RTr).

The system architect is then concerned only with the RTr, the internal system representation of the physical entity. Clearly, one issue is that this RTr is a discrete representation whose value becomes increasingly inconsistent, over time, with the real value of the entity. Therefore, to ensure correctness it is necessary that the representation error is kept sufficiently small, within a bound defined by the system designer. In other words, it is necessary to ensure that an RTr is *temporally consistent* with the RTe at all times.

In controlled and predictable environments it is usually possible to update the RTr periodically in order to solve the problem. However, this is not typically the case in networked, open and dynamic systems, or when the resources are limited. In these cases, it may be sufficient to be aware if the RTr is temporally consistent. To this end, the possibility of defining an interval during which the RTr will be temporally consistent is sufficient. This interval is called *temporal accuracy interval* for control [11] or *absolute validity interval* for databases [12].

In the context of networked sensor-based systems, there will be mechanisms to sense physical entities, process data and make it available for application use. The main objective is to perceive the surrounding environment and thus we call them *perception mechanisms*. The goal is to enrich these mechanisms with means to achieve *data validity awareness* or, better yet, secure some required *perception quality*.

We define data validity awareness as the possibility of knowing if the RTr is still valid. Then, we define perception quality as the error between the RTr and the RTe. The smaller the error, the higher the perception quality. Ideally, the application should be able to specify a desired perception quality, to be maintained throughout the operation, thus achieving temporal consistency. However, given the unpredictability characteristics of the considered environments, we believe that the best that can be done is estimating the achievable perception quality, not assuring it. As we discuss in Section VI, the possibility of providing a measure of the confidence, or coverage, of the assumed error, or perception quality, is one possible approach to achieve dependable perception.

IV. IMPEDIMENTS TO DATA VALIDITY AWARENESS AND PERCEPTION

From the above discussion it may seem that it is quite easy to achieve data validity awareness or even being able to dependably perceive the environment. However, this depends on the assumed environment, system and fault model. Therefore, in the following paragraphs we provide a brief systematization of the main issues, or impediments, that should be taken into consideration when addressing the problem.

We first look at the environment. It is fundamental to understand that the objective is to perceive the state of the environment, which is the state of physical dynamic entities. Clearly, a perceived value (a sample of the state) obtained at a certain moment can only be useful to infer the state at some later moment if the physical entity's dynamics is not arbitrary. Otherwise, a sample would only be useful to characterize the state at the precise moment the sample was obtained, in the past. The dynamics of the environment is thus an intrinsic impediment to dependable perception or validity awareness. To deal with that, it is necessary that assumptions concerning physical entities' behavior can be stated and will hold. Fortunately, this is the typical case, since at a macro-scale it is usually possible to assume that time-value entities like speed, temperature or distance vary sufficiently slow such that bounds exist on the maximum deviation between a sampled value (taken at some instant t) and the real value (at an instant $t + \Delta$). Nevertheless, care must be taken to ensure that assumptions are valid, that is, they have enough coverage in the concrete considered environment.

Regarding the system model, as we have been saying, we are interested in systems of uncertain synchrony, in which it is not possible to assume bounds on processing or communication delays, at least in general. In fact, at the system edges, in the sensor nodes, it may be reasonable to assume a real-time behavior. This makes it impossible to simply assume that real-time representatives in sink or other central processing nodes, i.e., the perceived state, can be always up to date. The system synchrony properties are thus an impediment to dependable perception. To deal with that, there are reasonable assumptions that can be made which restrict the asynchrony of the system. We mention three of such restrictions.

1) First, it may be possible to assume that the system is synchronous, with a high probability (or coverage), provided that sufficiently high bounds are considered. This is reasonable for applications that can live with these high bounds and accept the occasional violation of the assumption.
2) Second, asynchrony can be restricted by adding synchronized clocks to the system. There exist solutions,

namely based on external synchronization, which allow escaping the asynchrony of the network and thus limit the temporal gap between nodes in the system.

3) Finally, despite asynchrony, system timings are not arbitrary, but rather probabilistic. Asynchrony can thus be restricted if the probabilistic behavior of the system is known, or if it can be estimated from monitoring information. Differently from the firstly mentioned restriction, in this case the assumed bounds can vary during the execution and may be as good as allowed by the communication environment. They do not need to be too high if the environment is timely and will just be increased when synchrony degrades. On the other hand, this approach calls for adaptation capabilities, which may not always be implementable.

The last sort of impediments to dependable perception are faults in the system. Crash and omission faults prevent data from being collected or propagated in the system. Typically, the fault model must limit the number of such faults so that it may be possible to decide about the required degree of redundancy to deal with them. In networked sensor-based systems it is reasonable to assume the existence, or to build systems with multiple redundant sensor nodes and communication paths, such that these omissive faults can be handled. Regarding timing faults, they can occur if some of the above-mentioned restrictions to asynchrony are used to define a partially-synchronous model. In this case, it is important to carefully calculate the assumed bounds in order to reduce the probability of timing faults to acceptable levels.

Value faults can also considerably affect perception correctness, if not properly handled. In the distributed systems area, value faults tend to be disregarded, except when considering Byzantine environments, intrusions, or malicious nodes with the ability to change values. Otherwise, simple error correcting mechanisms ensure that accidental faults are handled and thus the fault model simply assumes that no value faults occur. In sensor-based systems the scenario is different. The process of acquiring the state of the environment, in sensors, is prone to several disturbing factors that may imply incorrect values being produced. In this case error correcting solutions are not applicable and therefore, this must be considered as an impediment to dependable perception. One way to deal with that is, once again, introducing redundant components. This is a typical solution that also makes use of sensor fusion techniques to sanitize, merge and process data. A less common solution implies understanding in detail the effects of disturbances on sensor produced values, and devise methods to specifically deal with each of those disturbances, for instance exploiting analytical models of the environment [13].

V. MEANS TO OBTAIN VALIDITY AWARENESS

Data validity awareness can be achieved with solutions based on explicit timing of events, which make use of clocks and timestamps, or solutions based on the implicit assumptions about timeliness or quality of service levels, exploited by perception mechanisms. In any case, a model of the environment dynamics is always necessary.

The definition of temporal validity intervals is an expedited way of raising awareness about data validity at any given moment. This can be a simple approach when synchronized clocks are available. In this case, sensor nodes can add a timestamp of the acquisition instant to each data value. Even if sampling and timestamping cannot be done atomically, the observation error can usually be bounded (handling value faults is an orthogonal issue). The validity interval can then be derived from the observed state, the environment model and the perception quality (allowed error) specified by the application. Any node having these three pieces of information, and by inspecting its local clock, is thus able to know if data is still valid.

Without synchronized clocks, an alternative is to do "time accounting" using only local time and making the first restriction mentioned above about the system synchrony. Sensed data must carry a time value corresponding to its "age", which is updated whenever it leaves a node and arrives at another. Local durations are measured using local clocks and upper bounds are considered for the transmission latency. In this way, a node can always know how much time has elapsed since the sensed data was obtained and thus become aware if it is still valid. Note that in this case the solution is implicitly assuming that some temporal guarantees exist.

Different solutions that are not based on temporal validity intervals can also be considered. They involve perception mechanisms, aimed at securing data validity and a desired perception quality, whose correctness depends on assumptions about the timeliness of the system, possibly expressed in terms of a quality of service specification. For example, these mechanisms can be like those used in typical real-time systems, with periodic sampling of the environment and periodic communication, which will work correctly (ensuring always valid data) if the assumed bounds are satisfied. To achieve validity awareness in these cases, the implicit assumptions made by the mechanisms must be tracked, which implies monitoring the execution. When the assumed QoS or timeliness is not satisfied in a system node, then state data dependent on these assumptions is deemed as invalid. To some extent, this can be seen as a form of failure detection, that also detects the violation of global assumptions.

Data validity awareness is important, particularly if applications can react when they detect that the perceived environment state is invalid. This is the case, for instance, of applications with a fail-safe state to which they can switch. However, a more ambitious objective is to be able to configure the system in order to control how well assumptions will be satisfied and hence data will be valid. Achieving such dependable perception is discussed in the next section.

VI. A DEPENDABLE PERCEPTION SERVICE

A dependable solution for environment perception must allow the system designer to know and to specify how well the application will behave in run time. Given that the system correctness rests on a number of assumptions that are made, in particular about the temporal behavior of the sensor network, achieving a dependable operation boils down to **securing the coverage** of these assumptions.

But how will a dependable perception service look like? At the interface the service will admit the specification of the required coverage, that is, the likelihood that it will provide a correct service. As the objective is to have a valid perception most of the time, the coverage will tend to be close to one, although this will depend on the specific application. Since resources are limited and it is not possible to control the synchrony of the sensor network, the trade-off is the perception quality. This trade-off may be exposed at the interface, letting the service user know the likelihood that a given or specified perception quality is achievable.

Central to this service will be a monitoring component, which will provide indications on the state of the network. Given previous research in other contexts [14], we believe that it is possible to characterize the stochastic behavior of a WSN based on its past behavior, and, making some reasonable assumptions about network stability, infer its near future behavior. The service will ensure that sensor data will be collected and transmitted whenever necessary, given the requirements, the knowledge about the network state and the environment model.

We now refer to some open issues concerning the implementation of such service. For instance, if we consider multi-hop networks, it may be useful to decompose end-to-end requirements on perception quality into partial specifications to drive the operation of intermediate nodes (which will have to run instances of the middleware). It may also be useful to integrate fault-tolerance measures with the dependable perception service, namely at the level of data-fusion mechanisms, using perception quality as an additional measure to consider in the fusion algorithms. Finally, the overhead introduced by all the necessary mechanisms must be evaluated and must be sufficiently small, particularly within sensor nodes. This is a particularly critical issue which must deserve considerable attention.

VII. Conclusion and Future Work

Ensuring dependable perception in networked sensor-based systems is a difficult problem. In this paper we discussed this problem, analyzing fundamental issues that need to be considered to devise adequate solutions and describing initial ideas to achieve these solutions.

We intend to pursue this work in two main directions. First, by assuming that communication delays are stochastic variables, we will extend previous work in [14] studying the concrete observed behaviors in wireless sensor networks, under varied operating conditions, and developing the probabilistic mechanisms for monitoring and for the estimation of temporal bounds, which will support dependable perception. Second, by developing middleware mechanisms for communication in WSNs, which will be driven by dependability requirements with respect to perception.

Acknowledgments

We wish to acknowledge Jörg Kaiser for the valuable discussions about some of the issues addressed in this paper. This work was partially supported by FCT through the Multiannual Funding and the CMU-Portugal Programs.

References

[1] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC protocols for wireless sensor networks: a survey," *Communications Magazine, IEEE*, vol. 44, no. 4, pp. 115–121, 2006.

[2] M. Chen, T. Kwon, and Y. Choi, "Energy-efficient Differentiated Directed Diffusion (EDDD) in Wireless Sensor Networks," *Comp. Comm.*, vol. 29, no. 2, pp. 231–245, 2006.

[3] P. Di Marco, P. Park, C. Fischione, and K. Johansson, "A dynamic energy-efficient protocol for reliable and timely communications for wireless sensor networks in control and automation," in *5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Networks - SECON 2009*, Rome, Italy, Jun. 2009.

[4] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He, "RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks," in *IEEE Real Time Technology and Applications Symposium*, 2002, pp. 55–66.

[5] V. C. Gungor, O. B. Akan, and I. F. Akyildiz, "A real-time and reliable transport (RT) 2 protocol for wireless sensor and actor networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 359–370, 2008.

[6] T. He, B. M. Blum, Q. Cao, J. A. Stankovic, S. H. Son, and T. F. Abdelzaher, "Robust and timely communication over highly dynamic sensor networks," *Real-Time Systems*, vol. 37, no. 3, pp. 261–289, 2007.

[7] S. Misra, M. Reisslein, and G. Xue, "A survey of multimedia streaming in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 1-4, pp. 18–39, 2008.

[8] M. Wang, J. Cao, J. Li, and S. K. Das, "Middleware for wireless sensor networks: A survey," *J. Comput. Sci. Technol.*, vol. 23, no. 3, pp. 305–326, 2008.

[9] E. Pignaton de Freitas, *A survey on adaptable middleware for wireless sensor networks*. Halmstad University, 2008.

[10] P. Veríssimo and L. Rodrigues, *Distributed Systems for System Architects*. Kluwer Academic Publishers, 2001.

[11] H. Kopetz, *Real-Time Systems*. Kluwer, 1997.

[12] K. Ramamritham, "The origin of TCs," in *Proceedings of the First ACM International Workshop on Active and Real-Time Database Systems*, Skovde, Sweden, Jun. 1995, pp. 50–62.

[13] S. Zug and J. Kaiser, "An approach towards smart fault-tolerant sensors," in *Proc. of the Int. Workshop on Robotics and Sensors Environments*, Lecco, Italy, Nov. 2010.

[14] M. Dixit, A. Casimiro, P. Lollini, A. Bondavalli, and P. Verissimo, "A probabilistic framework for automatic and dependable adaptation in dynamic environments," Dep. of Informatics, Univ. of Lisboa, Tech Report TR-09-19, 2009.