Dependable Perception in Wireless Sensor Networks

Luís Marques[1] and António Casimiro[2]

[1] FC/UL* `lmarques@lasige.di.fc.ul.pt`
[2] FC/UL `casim@di.fc.ul.pt`

Abstract. In this paper we present an analysis of perception dependability in wireless sensor networks. We put forward a model of perception and present the necessary definitions to understand and ascertain perception quality and dependability. We also present some directions for future work aiming at providing middleware support for perception dependability, using this perception model.

Abstract. Neste artigo apresentamos uma análise sobre confiabilidade na percepção em redes de sensores sem fios. Descrevemos um modelo de percepção e apresentamos as necessárias definições relativas à qualidade e confiabilidade da percepção. Indicamos também direcções de trabalho futuro com o objectivo de fornecer um middleware de supporte à percepção confiável, usando este modelo de percepção.

1 Introduction

Much work has been done in the last decade regarding distributed systems and applications based on networked sensors and actuators. Specifically, there has been great interest in wireless sensor networks (WSNs), in particular focusing on environment monitoring [1] and target tracking applications [2], through the use of small, low powered radio-enabled sensor nodes.

The general research area continues to garner attention, as can be seen through the current use of fashionable terms such as "cyber-physical systems" and "Internet of things", which refer to systems where information technology pervades and closely interacts with the real world environment. In fact, the real-time nature of information processed in these systems creates difficult problems, in particular if WSNs are to be used in sensor network-based control applications. Our present work is directed towards addressing some of these problems.

Despite the existence of a large body of work in this area, research has been limited to some specific topics, such as platforms and OS support, spacial coverage and awareness, synchronization, security, communication protocols and energy conservation (see [3] for a good survey). Overlooked in this research has been the issue of data quality, being either taken for granted or addressed with ad hoc validity mechanisms, mostly by using synchronized clocks [4] and setting validity deadlines for data, or simply by overwriting old data [5].

Lacking has been a systematic analysis of faults that may impair perception reliability and the definition of abstractions to allow reasoning in terms of perception quality. More practically, it is necessary to develop algorithms and mechanisms to deal with faults and improve perception quality and dependability. This can be done as part of a programming

* This work was partially supported by FCT through the Multiannual Funding and the CMU-Portugal Programs.

INForum 2010 - II Simpósio de Informática, Luís S. Barbosa, Miguel P. Correia (eds), 9-10 Setembro, 2010, pp. 715–718

model and middleware framework which would allow systems to be designed to explicitly reason about and be aware of perception quality and data validity.

In this paper we present a perception model upon which a middleware for dependable perception in WSNs can be built. We also present some possible directions for implementing such middleware.

2 Perception Model: Definitions and Dependability Issues

In our perception model we distinguish between environment entities and the systems' perception of them. We adopt the terminology in [6], referring to the former as *Real-Time entities* (RTe), which conceptually represent the actual state of some environment variables, exactly as they exist in a given moment, and to the latter as *Real-Time representatives* (RTr), which are the internal representation of a RTe, as imperfectly perceived by the system.

Having this RTe/RTr duality, we can thus define *perception quality* (or *observation error*) as the similitude (or difference) between a RTr and its respective RTe at some given time instant. The more similar a representative is to its RTe, the higher the perception quality.

Many applications require a minimum level of quality, so it is important to be able to distinguish between scenarios where an RTr is sufficiently similar to the corresponding RTe and scenarios where it is not. We can thus refer to *perception validity*, or *temporal consistency* of an RTr with respect to the RTe, which indicates if the perception quality is sufficient for an application. More formally, given a maximum allowed error ϵ, the perception is valid, or the RTr is temporally consistent at t, if $|\mathrm{RT}_r(t) - \mathrm{RT}_e(t)| \leq \epsilon$.

This definition of validity requires possessing knowledge of the real value of the RTe, and as such is suitable for theoretical reasoning, but cannot be directly used in practice (the state of the RTe is obviously unknown to the system). Instead, one or more of several different approximations can be used to derive the validity of an RTr. A suitable approach is to set a temporal validity interval or a deadline as soon as the RTe is observed (or sampled), after which the RTr value can no longer be considered valid. This approach requires a global notion of time in the distributed system and is dependant on the RTe's dynamics. Therefore, while it is always possible to perceive the state of any environment, doing so in a dependable way requires its dynamics to be somehow predictable, or no guarantee can be given about the validity of its perceived state. We thus consider as part of our perception model that we can bound the environment dynamics.

We consider that the perception of real-time entities can be affected in two ways. One is by faults and measurement errors affecting the sensing processes, such as malfunctioning transceivers and intrinsic sensor tolerance intervals. The other is by faults and temporal uncertainties affecting the communication in the wireless sensor network.

In this model we consider that faults are stochastic. We can then determine the best possible perception quality that is achievable with some probability, by considering the best-case sensing outcomes and the best-case transmission delays between information producers (sensing RTes) and consumers (using RTrs).

However, we need to distinguish the following two cases: either the system is able to eventually provide an RTr within the required error margin or it will never be able to do so. This second case would happen in situations where the fidelity of the sensors is insufficient for the required error margin or the transmission times are too large, always preventing updates of an RTr to be done in useful time.

If a system is able to eventually provide a valid RTr then the issue is that this validity cannot always be maintained, only being secured during some periods. In other words, it is possible to reason in terms of the probability that an RTr is valid at a given instant. We call this the *coverage* of an assumed error margin, which is a measure of the quality of service (QoS) provided by the WSN. In fact, we can generalize this concept so that completely unworkable QoS requirements are simply classified as being provided with zero coverage.

Redundancy can be used to deal with faults in WSNs and increase dependability. This can be spacial redundancy (the same information transmitted through different paths) or temporal redundancy (waiting more time to receive new updates, despite some lost ones). The impact of these approaches on the perception quality has to be investigated. For instance, waiting times in intermediate nodes during message propagation may be useful to overcome lost messages, but also imply a perception quality degradation.

Given the described model, middleware support can be provided to let applications enjoy a requested level of perception quality (ensure temporal consistency with a certain probability), or become aware of the maximum achievable quality. Providing awareness and probabilistic guarantees is a way to improve system dependability.

3 Middleware for Dependable Perception

Using the previously described perception model, a middleware for dependable perception can be built, which allows WSN-based applications to state their requirements concerning perception quality and coverage and have local real-time representatives be automatically updated accordingly. In the following paragraphs we sketch some general and preliminary ideas about what may be done by such distributed middleware.

The application specifies its requirements at a sink node. Then, before any actual sensor data dissemination occurs, the middleware undergoes a setup phase to propagate the requirements down the network to the relevant entities, i.e., active sensor nodes (which observe the RTe) and routing sensor nodes (which only propagate sensor data). A specification of the RTe dynamics must also be provided by the application at the sink node. When the network is configured, then the middleware enters a steady-state phase, during which sensor data is disseminated using a protocol that is designed to achieve the required quality/coverage specification of the RTr at the sink node. A key issue will be to manage the trade-off between the involved dissemination costs and the required dependability.

Depending on the nature of the considered RTe, there can be one or several active sensor nodes, performing actual sensing of the RTe (converting a physical state into its digital description). Every sensor node may also act as a routing node, propagating RTr updates to sink nodes. However, these different roles can be abstracted at the middleware level. Incoming data should be treated homogeneously, whether it has been received from a peer node, or has been generated locally by the sensing and transducing hardware. The task of the middleware is to decide when to propagate the data further and to which nodes. The decision will depend on several issues, including the dependability requirements and knowledge about the state of the network, failures, the number of hops to reach the sink and the actual validity of the received data. In essence, all the variables with impact on the validity of data should to be taken into account. In particular, this means that the middleware will also incorporate mechanisms to evaluate the operational context, which can have a simple local scope, or may involve distributed operations as well.

A possible functioning of the dependable perception middleware is then as follows. During the setup phase it is necessary to evaluate if the application request is feasible. By monitoring network conditions and using a fault probability model, the nodes are able to compute the amount of time that is necessary to ensure, with a given coverage, that RTr updates propagate between sensing nodes and sink nodes. If the middleware determines that a request is feasible then the sensor data dissemination mechanism can start. After the setup phase we can rely, with the specified coverage, on the RTr validity in the sink node. Preserving this coverage during the steady-state phase implies ensuring that each time the sink node's RTr is about to become invalid it should receive a RTr update which extends its validity. Intuitively, and without further concerns, sensor data generation and propagation should be as frequent and fast as possible, thus always ensuring the best possible RTr validity. However, this could be too resource consuming (in WSNs the objective is always to save resources) and even prejudicial to the objective of securing a given data validity, by creating network contention. Therefore, the middleware must incorporate mechanisms to evaluate the precise amount of resources that are needed to secure the application requirements, forwarding sensor data only when necessary and only through the strictly necessary redundant paths.

Given the dynamic and uncertain nature of the operational conditions, the potential need to accommodate topology changes and the heterogeneity of context perceived by each node, it is important that the middleware at each node continuously evaluates the amount of resources needed for data propagation. For instance, if no faults occur and an update is quickly propagated to an intermediate sensor, that sensor might recognize the slack still available for transmission and conclude that it can temporarily delay the propagation of the update. It could thus save energy if a subsequent update of the same RTr arrived at the node during that delay, thus improving the overall trade-off between validity and cost.

4 Conclusion

How to be aware and how to assure perception quality is still an open problem, relevant to many types of WSN applications. This paper discusses this problem and tries to provide some preliminary ideas on how to deal with it at the middleware level.

References

1. Tolle, G., Polastre, J., Szewczyk, R., Culler, D., Turner, N., Tu, K., Burgess, S., Dawson, T., Buonadonna, P., Gay, D., Hong, W.: A macroscope in the redwoods. In: SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems, ACM Press (2005) 51–63
2. Simon, G., Maróti, M., Lédeczi, A., Balogh, G., Kusy, B., Nádas, A., Pap, G., Sallai, J., Frampton, K.: Sensor network-based countersniper system, ACM Press (2004) 1–12
3. Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor network survey. Comput. Netw. **52**(12) (2008) 2292–2330
4. Rhee, I.K., Lee, J., Kim, J., Serpedin, E., Wu, Y.C.: Clock synchronization in wireless sensor networks: An overview. Sensors (2009) 56–85
5. Williamson, G., Cellai, D., Dobson, S., Nixon, P.: Modelling periodic data dissemination in wireless sensor networks. In: EMS '09: Proceedings of the 2009 Third UKSim European Symposium on Computer Modeling and Simulation, Washington, DC, USA, IEEE Computer Society (2009) 499–504
6. Veríssimo, P., Rodrigues, L.: Distributed Systems for System Architects. Kluwer Academic Publishers (2001)