# Towards Dependable and Stable Perception in Smart Environments with Timing and Value Faults

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.** Future physical environments are expected to be pervasively enriched with sensors, which mobile embedded applications can use to safely interact in and with that environment. Unfortunately, due to the open and uncertain nature of the environment and the wireless communication, it is not possible to provide strict a priori guarantees with regard to the quality and timeliness with which such environments can be perceived.
In this paper we take a look at the threats to a reliable perception of the environment, considering both timing and value faults. We discuss how such threats can be mitigated and we explore possible paths towards an integrated architecture to efficiently achieve a dependable and stable perception of smart environments in the presence of timing and value faults.

**Keywords:** smart environments; dependability; adaptation; stability; real-time; fault-tolerance;

## 1 Introduction

For more than a decade now, there has been a significant interest in the area of distributed sensors which communicate through wireless networking. This is reflected in such concepts as "Cyber-physical Systems", "Internet of Things", "Wireless Sensor Networks" (WSNs) and "Smart Environments". What is common to all of these concepts is the vision of highly pervasive sensors which allow the environment to be monitored at large scales, through the cooperation of many individual systems.

This mesh of highly ubiquitous sensors can support a wide range of different applications, some of which have already received significant attention, such as habitat monitoring [1][2], object tracking [3], target tracking [4], detection of pollutants [5], climate monitoring [6], energy consumption awareness [7], early disaster warning systems [8], and smart vehicles [9].

* This work was partially supported by the EU through the KARYON project (FP7-288195) and the FCT through the Multiannual Funding Program.

Different applications have different requirements with regard to the accuracy and timeliness with which the physical environment must be perceived. For instance, a pollutant monitoring application may only require that gas concentrations be measured every few minutes or hours, and it may be that this data does not need to be collected in real-time. Furthermore, the data may be able to be collected and processed in a centralized fashion, allowing for sensor readings to be compared and correlated in a batch and non-real-time manner, such as to remove outlier values due to faulty sensors.

On the other hand, other applications can have strict accuracy and timeliness requirements. For instance, target tracking and smart vehicles operating as part of vehicular networks are examples of applications which require both a highly accurate and a timely perception of the environment, to assure operational safety. In general, mobile and cooperative systems, that cooperate in and with the physical environment, will require stricter guarantees than more "passive" systems, such as habitat monitoring.

Yet, the problem arises that in open and uncertain environments it is not possible to provide a priori guarantees regarding the accuracy and timeliness with which the information from the surrounding environment and other cooperative systems can be obtained. Without such guarantees it becomes harder for mobile systems, such as vehicles, to safely cooperate and interact.

Part of the problem stems from the way in which embedded systems have, traditionally, been designed, which is not compatible with such open environments. Traditionally, embedded systems — particularly safety-critical systems — have been engineered in a systematic way which allows providing the required properties in terms of timeliness, accuracy and validity of the sensed environment.

To provide timeliness guarantees, all scenarios are previously studied, all operations are given deadlines, critical communication is performed using real-time networks, and every task is scheduled using carefully crafted algorithms, which execute code paths with worst case execution times designed to respect such deadlines. To provide the necessary accuracy, the type and amount of sensors used is carefully planned at system design time. The necessary redundancy is built-in, to provide protection against both timing and value faults.

Since traditional systems are engineered in such a way that all requirements are assured by design, their architectures have generally not focused on being aware, at runtime, of how well system assumptions are being covered, nor do they generally exhibit a large degree of runtime adaptability.

We argue that in open environments, in which the environment state information is obtained with uncertain timeliness and accuracy, such systems must be engineered to be adaptable and aware of their operational environment. We must transition from hard limits to adaptable bounds, from no awareness of the operating environment conditions to an awareness of runtime timing and accuracy bounds, and from uncertainty to dependable perception and operation.

We previously proposed a generic technique to be aware of, and to dependably adapt to, the varying communication timings of WSNs [10]. Such tech-

nique, based on a stochastic analysis of the runtime latencies of the wireless network, allows one to overcome the communication uncertainties of these open environments. In fact, we evaluated the proposed technique and confirmed its dependability in 802.15.4-based WSNs [11].

In this paper we discuss how similar techniques can incorporate the presence of sensors with varying margins of error, sensor heterogeneity and value faults, and how to achieve adaptation stability. We look at the various threats to a reliable perception of the environment, discuss how they can be mitigated and how to incorporate such strategies in an efficient architecture to achieve a stable and dependable perception of the environments, even in the presence of timing and value faults.

In the following section we start by exploring the desired properties of smart environments and various threats to a reliable perception.

## 2 Threats to Reliable Perception

We can identify several properties of smart environments upon which a dependable perception of the environment relies. With regard to the values which represent the current state of the perceived physical environment, or the production thereof, the following are four important properties.

- **Accuracy** — reflects the expected or computed margin of error of the considered value;
- **Validity** — indicates if the value may be outside of the expected or computed margin of error, due to a value fault;
- **Timeliness** — the ability to produce and deliver the desired value within a given deadline;
- **Efficiency** — the relation between the amount of resources expended and the accuracy, validity and timeliness of the obtained value.

These properties are not completely orthogonal but, as defined here, are useful constructs to allow reasoning about the threats to a reliable perception of the environment.

The properties of accuracy, validity and timeliness will determine the perception reliability and, therefore, the **application performance**. The relationship between these properties of the smart environment and the obtained performance is necessarily application-specific. In open environments these properties will also vary throughout time and space. As such, the reliability of the environment perception, and consequent application performance, will depend on: (1) the performance with which the applications or the smart environment itself adapt to varying conditions; (2) the stability of the three properties; (3) the relationship between (1) and (2).

Therefore, we must not only be aware of current environment conditions, but also of how such conditions can vary throughout time and space. Only that way can the application or the smart environment avoid adapting to new conditions more frequently (or faster) than would be optimal for a given application.

Based on this understanding of the requirements to a reliable perception of the environment, we can identify several threats and challenges to this reliability.

- **The amount of information** — with the amount of sensors spread out throughout the environment expected to grow exponentially, there will be added pressure on scarce resources like bandwidth (which will likely grow slower than the amount of information) and computational power (which is probably even more crucial in small and limited sensors);
- **Sensor heterogeneity** — the availability of sensors which provide the same state information with different accuracies, or which provide complementary state information, is an opportunity for optimization, but can also impair perception quality if the best information sources are not chosen;
- **The variability of information** — for mobile systems, the geographical variability of sensor information can have an impact on performance, by jeopardizing adaptation stability;
- **CPU hardware / software faults** — computational faults in the systems comprising either the smart environment itself or the cooperative systems can compromise safety, by corrupting sensor information or coordination information;
- **Sensor transducer faults** — in a world with a very large number of sensors, possibly from many different manufacturers, it can be expected that faulty sensor readings are a common occurrence, in absolute terms. The faults can derive from transducer miscalibrations, transducer aging, environmental effects, electromagnetic interferences, and various other sources;
- **Communication interferences** — both intra-network interferences (network nodes competing for the transmission medium) and external interferences (e.g. background noise) can threaten reliable environment perception, essentially by means of timing failures. Value faults can also occur, through packet corruption, but those faults can be transformed into omissions;
- **Network inaccessibility** — communication interferences may also lead the network to temporarily refrain from providing service, even if it is not considered to have failed. Network inaccessibility can be characterized by the specification of limits for inaccessibility duration and rate, where the violation of those limits implies a permanent failure of the network.
- **Clock desynchronization** — clock drift and other sources of desynchronization can impact distributed sensors, when relying on a notion of global time;

## 3 Accurate and Timely Perception

The reliability of the environment perception and consequent application performance will depend not only on the accuracy and timeliness of the sensed information, but also on the stability of these metrics. In this section we discuss existing work to achieve value accuracy, fault tolerance and timeliness, and to what extent these techniques can provide the required properties in a stable way, or be augmented to do so.

### 3.1 Timeliness

In the work described in [10], we previously proposed a technique to achieve probabilistic timely behavior in WSNs. A central assumption of that work was that although the timing variables had unknown bounds, which in addition could change at any time due to the open nature of the environment, such bounds were not completely arbitrary and unpredictable but that, instead, they were probabilistic. Furthermore, we considered that these probabilistic bounds changed slowly enough compared with the capacity of the application or the WSN itself to recognize and adapt to new bounds.

In [11] we indirectly evaluated these assumptions in 802.15.4-based WSNs, by measuring the adaptation effectiveness with regard to the end-to-end latencies, under a variety of scenarios. We concluded that, even with the adaptation occurring only at the application level (and not at the level of the WSN itself), we could meet deadlines with the desired probability, plus or minus a small margin of error (generally 1 or 2%, with some scenarios having a maximum margin of slightly over 5%).

There are three important issues to adaptation stability which this work did not cover. One is that these deadline fulfillment metrics are long-term averages; the work in [11] did not focus on the short-term variability. A second is that it was assumed that an application could adapt instantly after the new network behavior was recognized. Yet a third is that we had devised this technique mostly with stationary WSNs in mind.

Regarding this last point, notwithstanding our initial assumptions, in our evaluation we verified that the proposed technique was effective not only in stationary networks but also when sensor mobility was introduced. Despite the significant additional dynamics, we observed only a small increase in the margin of error of the probability of deadlines being fulfilled. In networks with more spatial heterogeneity and/or faster node movement it might be necessary to take proactive measures to assure that adaptation stability remains.

Regarding the short-term stability of the deadline fulfillment probability, although this was not specifically tested we informally observed that there were no significant variations from period to period. If stronger guarantees are needed, one option that is likely viable is to dynamically change the WSN behavior to adjust at runtime the amount of resources expended, to decrease variability. The challenge is how to do this in a decentralized and efficient way.

Likewise, while applications may not be able to adapt instantly, especially if they are far away from the source of network disruption, having the adaptation occur at the level of the WSN will likely be an effective strategy to assure temporal stability of timing bounds.

In fact, one of our conclusions from our evaluation in [11] supports the effectiveness of this strategy. We observed that in more complex networks, with more sources of uncertainty, we achieved (seemingly paradoxically) *better* adaptation effectiveness than in simpler networks. The reason for this is that all of these sources of uncertainty can average out. Hence, in future smart environments, with large amounts of sensors and network nodes, we can expect that decentral-

ized network-level adaptations will be highly effective in smoothing out short term timing variabilities.

### 3.2 Accuracy and Value Faults

In traditional embedded systems, the type and number of sensors are predetermined according to the application requirements, and therefore fault masking can be planned in a straightforward manner using the sensors' manufacturer specification sheet. In smart environments the amount and type of available sensors is unknown at design time and can vary unpredictably. Therefore, the accuracy and fault behavior cannot be assumed in such a black box style, but must instead be checked and enforced at runtime.

An architecture for a dependable distributed sensor system is described in [12], which allows for an efficient detection and masking of common types of sensor faults. This architecture integrates a set of ideas that were previously developed in isolation into a coherent and unifying concept. We here review some of the underlying concepts and its suitability to be extended into an architecture for accurate, timely and stable environment perception.

One strategy to implement fault detection is through the classical paradigm of hardware replication. The work presented in [13] had previously identified the necessary number of sensors to tolerate different types of transducer faults; a different method was described in [14], based on maximizing the consistency of sensor fusion results.

Another strategy is to detect anomalous values from a single sensor, by comparing the sensor output to a model of the system and noticing discrepancies, or through a signal analysis of the sensor output; different approaches to implement this are detailed in [15], [16] and [17].

The distributed sensor architecture proposed in [12] combines ideas from these different approaches, by performing each kind of fault checking as close as possible to the source of error, for efficiency. This architecture follows the model of distributed fusion architecture, which had already been developed but which generally do not consider fault tolerance [18], extending previous work on fault-tolerant sensors [19].

This architecture performs a series of tests and, in the end, outputs sensor values together with a measure of their validity, which is a computed probability of the respective value being faulty. This final validity is a combination of the computed probabilities for each of the possible fault types.

One important aspect of this architecture is that it seems to be compatible with the technique proposed in [10] to achieve probabilistic real-time guarantees. For instance, regarding the distributed sensor fusion, the architecture does not specify any particular protocol for the dissemination and aggregation of sensor values, and therefore does not introduce incompatibilities with real-time network protocols. Also, no specific algorithms are mandated for the local transducer fault checks, so the algorithms can be chosen considering on their impact on timeliness, for example based on their worst-case execution times.

The architecture considers that the accuracy and fault model of the sensors is discoverable but does not contribute mechanisms to assure the provision of specific accuracy or fault probabilities. There are two considerations here for adaptation stability. One is that, even in stationary networks, the identified fault probability can (and will likely) vary throughout time. This can happen either because of changes in the sensor themselves or because of variations in the amount and type of aggregated sensor information. The other is that, in scenarios of mobility, such as vehicular scenarios, the quality of sensor information will likely vary spatially.

### 3.3 Adaptation Stability and Application Performance

We consider that an adaptation is stable for a given application if another adaptation will *not* be required, with a given probability $p$, in the immediate interval $\delta$ after it becomes effective. The values of $p$ and $\delta$ will vary with the considered application and desired application performance.

In this subsection we clarify the impact of the adaptation stability in the application performance through a hypothetical scenario.

Figure 1 illustrates a scenario with moving vehicles and heterogenous perception quality regions. Vehicle A is shown in its current position (black) and a past position (grey).

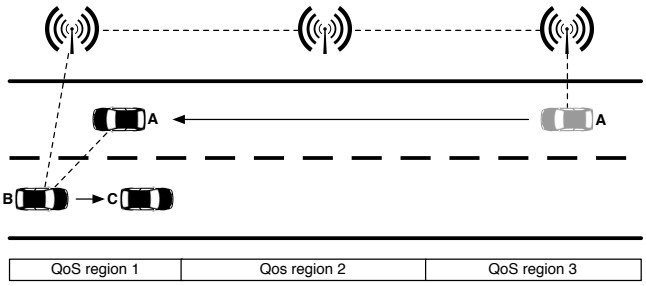


Fig. 1: Timing Variables (Example)

The operational performance of vehicle B, for instance with regard to the distance that must be kept to vehicle C, will be depend on the perception quality and stability that is achievable in QoS region 1. Since vehicles cannot instantaneously change their speed, and since frequent speed changes have a cost in terms of fuel consumption and passenger comfort, the optimal distance will depend on the expected probability of receiving within a given period sensory updates that are accurate and valid. If such information is not received then the operational performance will have to be degraded, if necessary to the point of switching to a fail-safe mode (e.g. vehicle immobilization).

Due to the spatial QoS heterogeneity, as vehicles move they will have to adapt to maintain operational safety and performance. If a more global awareness of perception quality is available then this adaptation can be optimized to maximize application performance. For instance, if QoS region 3 has a lower quality and that information is available in QoS region 1 then the vehicles can preemptively and gradually transition to the new optimal state, as they approach the region with lower quality. In particular, an adaptation can be made at QoS region 1 or 2 which is stable enough that, with high probability, no further adaptation is required when the vehicle reaches QoS region 3.

A possible way to achieve such awareness is for the smart environment participants to disseminate their local awareness of perception quality and temporal stability thereof. With regard to the timeliness, the historical latencies that are collected already contain information about the temporal variability of network latencies. In terms of sensor accuracy and validity, the distributed sensor fusion architecture should be extended to collect historical information and to compute the probability of valid updates being received within a period.

There are several challenges to an effective and efficient dissemination of the achievable perception quality information. For instance, in Figure 1 we can observe that the perception quality achievable in region 3 can be disseminated to other vehicles either directly by vehicle A or indirectly, by using the road-side infrastructure. Which is the best choice can depend on a wide variety of factors. If bandwidth is particularly scarce, then it might be more efficient for vehicle A to directly communicate with vehicle B. If vehicle A is moving slowly, then it might be preferable for the information to be disseminated through the road-side infrastructure, so that it is available in a timely manner, before vehicle B is too close to region 3. If multiple applications can benefit from that information, then it might be preferable to use the road-side infrastructure as a central hub of dissemination. Also, we can expect vehicles to host and support specific applications, while a smart environment sensor infrastructure can be expected to be more general purpose. Therefore, while vehicles might be preprogrammed to disseminate the quality of certain specific environment attributes, the generic environment sensors will have to learn what information is most beneficial to surrounding applications.

A more specific consideration is how to predict future sensor validities based on the disseminated past sensor information and/or validities. The non-parametric approach, based on order statistics, used in [10] may not be (fully) applicable to this task. It will be necessary to evaluate what model best approximates the variations over time of such validities. This model will have to integrate an awareness of both (1) how likely a fault is to occur in a specific sensor value, given past faults, and (2) how likely a valid value is to be available until a given deadline, given what fusible sensor values are expected to arrive.

## 4 Towards an Architecture for Dependable and Stable Environment Perception

Although no specific architecture is proposed in this paper, from the issues previously examined we can start enumerating components, services, design aspects and guiding principles that should comprise an architecture for dependable and stable environment perception.

In general, we suggest an architecture which uses fault-tolerant distributed sensor fusion to handle sensor accuracy and faults, that has an awareness of network latencies, and where the communication deadlines can be probabilistically assured by dynamically varying the amount of resources expended by the networked sensors of the smart environment, to counteract network and application dynamics.

The CPU hardware and/or software faults which change sensor values can be dealt with by enforcing fail-silent behavior. Other techniques can be compared for efficiency and performance in different scenarios, such as value voting and the elimination of outliers.

The threat introduced by network inaccessibility can be dealt with using specialized techniques, as has already been done for other kinds of networks [20][21]. These techniques can be compared for efficiency and performance with using redundant networks paths, which are unlikely to suffer of inaccessibility at the exact same times.

Clock synchronization is assumed in both the probabilistic timeliness solution and in the fault-tolerant distributed sensor fusion architecture. Such clock synchronization can be achieved by using algorithms specially optimized for smart environments. There already exist algorithms optimized for WSNs [22][23]. These can be improved upon to take into consideration factors that would be specific to an architecture for smart environments. For instance, synchronization accuracy may not need to be homogenous; it may be more efficient for the accuracy to be relatively better between nodes which are more relevant for sensor fusion, which may rely on the transformation of sensed values according to the elapsed time, as considered in [12]. Also, an awareness of network latencies may be exploited for a more efficient synchronization algorithm, at no additional cost if this information is already collected to assure timeliness.

There are many ways to mitigate network interferences [24]. To the extent that such interferences are not unpredictable they will not have an impact on timeliness, when using the considered probabilistic approach, but instead will only affect the efficiency of the environment perception. Strategies for improving that efficiency must not jeopardize timing properties. With regard to fault tolerance, the architecture can integrate the provision of timeliness properties with sensor fusion, so that guarantees of accuracy and validity will not be threatened by the necessary sensor information not being available in a timely manner.

A central aspect of this architecture is the components, protocols and/or services for the discovery of sensor information, which should work in such a way that the most beneficial combinations of fusible sensor state are made available. Another fundamental aspect, as identified in section 3.3, is how to provide stable

perception quality properties to applications, throughout not only time but also space.

We propose exploring architectural mechanisms that can efficiently satisfy both of these aspects. In particular, we propose researching a mechanism for the decentralized dissemination of both the available sensors and their fault models, as well as what validity can be achieved in a given spatial region, according to the combination of perceived network latency conditions, available sensors and historical sensor validities.

## 5 Conclusion

The trend is clear that an ever increasing amount of sensors will be part of the physical environment, eventually culminating in smart environments where sensors are pervasive. While this presents an opportunity for increased autonomy and performance of mobile applications, the lack of guarantees offered by these environments creates a very hard challenge for application dependability and, in particular, their operational safety.

We identified the main threats to dependability and presented various possibilities of how both accuracy and timeliness might be achievable in an architecture for a dependable and stable environment perception. We built upon previous work, and explored how to combine and extend one solution devised to provide probabilistic timeline guarantees and an architecture for fault-tolerant distributed sensor fusion.

We identified the limitations of these previous efforts in terms of perception/performance stability, and we suggested various research venues of how such stability might be supplemented.

## 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. Szewczyk, R., Osterweil, E., Polastre, J., Hamilton, M., Mainwaring, A., Estrin, D.: Habitat monitoring with sensor networks. Communications of the ACM **47** (2004) 34–40
3. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The Cricket Location-Support System. In: 6th ACM MOBICOM, Boston, MA (August 2000)
4. 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
5. Tsujita, W., Yoshino, A., Ishida, H., Moriizumi, T.: Gas sensor network for air-pollution monitoring. Sensors and Actuators B: Chemical **110**(2) (October 2005) 304–311
6. Leonard, N.E., Paley, D., Lekien, F., Sepulchre, R., Fratantoni, D., Davis, R.: Collective motion, sensor networks, and ocean sampling. Proceedings of the IEEE,

special issue on the emerging technology of networked control systems (95) (2007) 48–74
7. Jiang, X., Dawson-Haggerty, S., Dutta, P., Culler, D.: Design and implementation of a high-fidelity ac metering network. In: Proceedings of the 2009 International Conference on Information Processing in Sensor Networks. IPSN '09, Washington, DC, USA, IEEE Computer Society (2009) 253–264
8. Basha, E.A., Ravela, S., Rus, D.: Model-based monitoring for early warning flood detection. In: Proceedings of the 6th ACM conference on Embedded network sensor systems. SenSys '08, New York, NY, USA, ACM (2008) 295–308
9. Lee, U., Magistretti, E., Zhou, B., Gerla, M., Bellavista, P., Corradi, A.: Mobeyes: Smart mobs for urban monitoring with vehicular sensor networks. IEEE Wireless Communications Vol. 13, No. 5 (2006)
10. Marques, L., Casimiro, A.: Lightweight dependable adaptation for wireless sensor networks. In: Proceedings of the 30th IEEE International Symposium on Reliable Distributed Systems Workshops, 4th International Workshop on Dependable Network Computing and Mobile Systems (DNCMS 2011), Madrid, Spain
11. Marques, L., Casimiro, A.: Evaluating lightweight dependable adaptation in 802.15.4 wireless sensor networks. Technical report, TR-2012-04, Dep. of Informatics, Univ. of Lisboa. http://docs.di.fc.ul.pt/handle/10455/6873
12. Zug, S., Dietrich, A., Kaiser, J.: An architecture for a dependable distributed sensor system. IEEE T. Instrumentation and Measurement **60**(2) (2011) 408–419
13. Chen, C., Brown, D., Sconyers, C., Zhang, B., Vachtsevanos, G., Orchard, M.E.: An integrated architecture for fault diagnosis and failure prognosis of complex engineering systems. Expert Syst. Appl. **39**(10) (August 2012) 9031–9040
14. Marzullo, K.: Tolerating failures of continuous-valued sensors. ACM Trans. Comput. Syst. **8**(4) (November 1990) 284–304
15. Koushanfar, F., Potkonjak, M., Sangiovanni-Vincentelli, A.: On-line fault detection of sensor measurements. In: IEEE Sensors. (2003) 974–980
16. Isermann, R.: Model-based fault detection and diagnosis: status and applications. In: In Proceedings of the 16th IFAC Symposium on Automatic Control in Aerospace, St. (2004) 71–85
17. Doebling, S.W., Farrar, C.R., Prime, M.B.: A summary review of vibration-based damage identification methods. Identification Methods," The Shock and Vibration Digest **30** (1998) 91–105
18. Makarenko, A., Durrant-whyte, H.: Decentralized data fusion and control in active sensor networks. In: In Proceedings of the Seventh International Conference on Information Fusion. (2004)
19. Zug, S., Kaiser, J.: An approach towards smart fault-tolerant sensors. In: Proceedings of IEEE International Workshop on Robotic and Sensors Environments (ROSE2009), Lecco, Italy (November 2009)
20. Rufino, J., Veríssimo, P., Almeida, C., Arroz, G.: Integrating inaccessibility control and timer management in canely. In: ETFA, IEEE (2006) 348–355
21. Souza, J.L.R., Rufino, J.: An approach to enhance the timeliness of wireless communications. In: The Fifth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM), Lisbon (November 2011)
22. Li, Q., Rus, D.: Global clock synchronization in sensor networks. IEEE Transactions on Computers **55**(2) (2006) 214–226
23. Yoon, S., Veerarittiphan, C., Sichitiu, M.L.: Tiny-sync: Tight time synchronization for wireless sensor networks. ACM Trans. Sen. Netw. **3**(2) (June 2007)
24. Liang, C.J.M.: Interference characterization and mitigation in large-scale wireless sensor networks (2011)