Research Roadmap on Cooperating Objects

Pedro José Marrón, Stamatis Karnouskos, Daniel Minder
and the CONET consortium

CONET
COOPERATING OBJECTS
NETWORK OF EXCELLENCE

RESEARCH ROADMAP ON COOPERATING OBJECTS

KK-78-09-698-EN-C

ISBN 978-92-79-12046-6

9 789279 120466

Publications Office

Research Roadmap
on Cooperating Objects

Pedro José Marrón, Stamatis Karnouskos, Daniel Minder
and the CONET consortium

Cataloguing data can be found at the end of this publication.

Luxembourg: Office for Official Publications of the European Communities, 2009

ISBN 978-92-79-12046-6
DOI 10.2759/11566

Printed in Luxembourg

Printed on white chlorine-free paper

CONET Project Coordinator

Prof. Dr. Pedro José Marrón

University of Bonn and Fraunhofer IAIS
Institute of Computer Science IV
Roemerstr. 164
D-53117 Bonn
Germany

Tel: +49-228-73-4220
Fax: +49-228-73-4508
E-mail: pjmarron@cs.uni-bonn.de
http://iv.cs.uni-bonn.de
http://www.cooperating-objects.eu

CONET Roadmap Responsible

Stamatis Karnouskos

SAP Research
SAP AG
Vincenz-Priessnitz-Strasse 1
D-76131 Karlsruhe
Germany

Tel: +49-6227-752515
Fax: +49-6227-7845799
Email: stamatis.karnouskos@sap.com
http://www.sap.com

Responsible EC Project Officer

Dr. Jorge Pereira
Principal Scientific Officer
European Commission
DG Information Society and Media

Office BU31 05/21
B-1149 Brussels
Belgium

Tel: +32-2-296-1547
Fax: +32-2-299-4353
Email: Jorge.Pereira@ec.europa.eu

The CONET consortium

universität bonn

TELECOM ITALIA

TU Delft

SELEX GALILEO

isep Instituto Superior de Engenharia do Porto

SICS

TU berlin

Schneider Electric

aicia

BOEING

NUI Galway
OÉ Gaillimh

IN SUPREMÆ DIGNITATIS 1343
UNIVERSITÀ DI PISA

UCL

University of Cyprus

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

SAP®

The work of creating this roadmap was carried out as part of activities performed within the European Project CONET, the Cooperating Objects Network of Excellence funded by the European Commission between June 1^{st}, 2008 and May 31^{st}, 2012 under contract number ICT-2007-2-224053. We would like to extend our deepest gratitude to the following individuals that have contributed to writing this document:

Rheinische Friedrich-Wilhelms-Universität Bonn (UBONN), Coordinator

- Nils Aschenbruck
- Matthias Gauger
- Marcus Handte
- Muhammad Haroon
- Muhammad Umer Iqbal
- Pedro José Marrón
- Daniel Minder
- Robert Sauter
- Chia Yen Shih

Asociación de Investigación y Cooperación Industrial de Andalucía (AICIA)

- Fernando Caballero
- Sabine Delaitre
- Adrián Jiménez
- Iván Maza
- Aníbal Ollero
- José Manuel Sánchez Matamoros
- J. Ramiro Martínez-de-Dios

Instituto Superior de Engenharia do Porto - Instituto Politécnico do Porto (ISEP)

- Mário Alves
- Björn Andersson
- Nouha Baccour
- Anis Koubâa
- Ricardo Severino
- Paulo Gandra de Sousa

Swedish Institute of Computer Science (SICS)

- Adam Dunkels
- Luca Mottola
- Thiemo Voigt

Eidgenössische Technische Hochschule Zürich (ETHZ)

- Kay Römer

National University of Ireland, Galway (NUIG)

- Manfred Hauswirth
- Vinny Reynolds
- Marco Zúñiga

Technische Universität Berlin (TUB)

- Onur Ergin
- Vlado Handziski
- Andreas Köpke
- Thomas Mentzel
- Andreas Willig
- Daniel Willkomm

University of Cyprus (UCY)

- George Samaras
- Panayiotis Andreou

Università di Pisa (UNIPI)

- Antonio Bicchi
- Gianluca Dini
- Adriano Fagiolini
- Lucia Pallottino
- Ida Maria Savino
- Riccardo Schiavi

Technical University Delft (TUD)

- Koen Langendoen
- Otto Visser

University College London (UCL)

- Yang Yang

SELEX Galileo (SELEX)

- Graham Bourdon
- George Matich

SAP Research (SAP)

- Stamatis Karnouskos

Boeing Research and Technology Europe (BRTE)

- David Scarlatti
- Sudhakar S. Shetty
- Ignacio Zozaya

Schneider Electric (SCHNEIDER)

- Armando Walter Colombo
- Nataliya Popova

Telecom Italia (TI)

- Fabio Bellifemine
- Claudio Borean
- Alberto Cuda
- Roberto de Bonis
- Sinem Ergen
- Massimo Maggiorotti
- Marco Sgroi

Universidad Pablo de Olavide (UPO), associated member

- Luis Merino

Several people have provided valuable feedback to our roadmap either directly reviewing the intermediate versions of the document or via the surveys that the CONET Consortium has conducted. We would like to thank the anonymous participants in the roadmap surveys and the following people that kindly provided us with their name and affiliation:

Marco Aiello	University of Groningen
Ernesto A. Arzabala Contreras	Instituto Tecnologico de Chihuahua
Marcel Baunach	University of Würzburg
Jan Beutel	ETH Zürich
Tim Denis	52M
Jean-Dominique Decotignie	Centre Suisse d'Electronique et de Microtechnique (CSEM)
Jarek Domaszewicz	Warsaw University of Technology
Martin Elixmann	Philips
Georg Gaderer	Austrian Academy of Science
Anders Isvén	Excellent Growth Methods (EXGM)
Martin Klepal	Cork Institute of Technology
Costis Kompis	VODERA Ltd.
Ioannis Krontiris	University of Mannheim
Spyros Lalis	University of Thessaly
Lars-Åke Larzon	Uppsala University
Gaby Lenhart	European Telecommunications Standards Institute (ETSI)
Patrick Loschmidt	Austrian Academy of Science
Marco Martalo	University of Parma
Paolo Medagliani	University of Parma
Clemens Mühlberger	University of Würzburg
Ioannis Papaefstathiou	Telecommunication Systems Institute
Paolo Proietti	ELSAG DATAMAT spa
Davide Quaglia	University of Verona / EDALAB S.R.L.
Utz Roedig	Lancaster University
Vinay Sachidananda	TU Darmstadt
Albert Ali Salah	Centrum Wiskunde & Informatica (CWI)
Manolo Serrano	ETRA Investigación y Desarrollo, S.A
Markus Taumberger	VTT Technical Research Centre of Finland
Wei Wang	ARM
Koen Williame	52M

Contents

List of Figures

List of Tables

Chapter 1

Executive Summary

The field of Cooperating Objects envisions vast numbers of embedded devices, such as networks of sensors and actuators, industrial production lines and machines, and household appliances that are interconnected and cooperate with each other in order to provide advance services. The functionality and sensor data these devices will be offering, are often referred as *real-world services* because they are provided by embedded devices, which are part of the physical world. Unlike most traditional enterprise services, which are designed to interact with human users, real-world services provide real-time data about the physical world.

According to several market studies, the number of devices around us is going to continue to grow tremendously and, more significantly, these devices will not be isolated! As the advances on networked embedded devices have been overwhelming, these devices will be able to communicate with each other and develop cooperation capabilities.

A study from ON World Inc.[176] projects that Wireless Sensor Network (WSN) systems and services will be worth $6.6 Bn in 2011, and in 2012 it is expected that $25.1 million WSN units will be sold for smart home solutions only, a significant increase from the 2 million in 2007. The business opportunities for real-world services are huge [437]. As mass market penetration of networked embedded devices is realized, services taking advantage of the novel functionality of devices will give birth to new innovative applications and provide both revenue generating and cost saving business advantages. From a technological point of view, the key challenge is how to discover, assess, and efficiently integrate the new data points into business applications.

Wireless Sensor Networks are a canonical example of a wider field dealing with Cooperating Objects that attempts to create the necessary technologies to make the vision of Mark Weiser of the disappearing computer a reality. Cooperating Objects are, in the most general case, small computing devices equipped with wireless communication capabilities that are able to cooperate and organize themselves autonomously into networks of sensors, actuators and processing units to achieve a common task.

The book you have in your hands contains information about the research roadmap envisioned for Cooperating Objects by the CONET consortium (www.cooperating-objects.eu) and its associated industrial and academic partners. The project was started in June 2008 as a Network of Excellence co-funded by the European Commission to investigate the field of Cooperating Objects, foster the collaboration of experts already doing research in this field as part of a network of excellence, and devise a research roadmap that could be used for the definition of future research programs within the European Commission. This book is the result of the first year of the project and, although it is based on a previous research roadmap created as part of the Embedded WiSeNts project (www.embedded-wisents.org), the roadmap has been significantly expanded to include the latest advances in the field.

1.1 Definition of Cooperating Objects

A number of different system concepts have become apparent in the broader context of embedded systems over the past couple of years. First, there is the classic concept of **embedded systems** as mainly a control system for some physical process (machinery, automobiles, etc.). More recently, the notion of pervasive and **ubiquitous computing** started to evolve, where objects of everyday use can be endowed with some form of computational capacity, and perhaps with some simple sensing and communication facilities. However, most recently, the idea of **wireless sensor networks** has started to appear, where entities that sense their environment not only operate individually, but collaborate together using ad-hoc network technologies to achieve a well-defined purpose of supervision/monitoring of some area, some particular process, etc.

We claim that these three types of systems that act and react on their environment are actually quite diverse, novel systems that, on the one hand, share some principal commonalities and, on the other hand, have some different aspects that complement each other to form a coherent group of objects that cooperate with each other to interact with their environment. In particular, important notions such as control, heterogeneity, wireless communication, dynamics/ad-hoc nature, and cost are present to various degrees in each of these types of systems.

The conception of a future-proof system would have to combine the strong points of all three system concepts at least in the following functional aspects:

- Support the control of physical processes in a similar way embedded systems are able to do today.
- Have as good support for device heterogeneity and spontaneity of usage as pervasive and ubiquitous computing approaches have today.
- Be as cost efficient and versatile in terms of the use of wireless technology as Wireless Sensor Networks are.

For these reasons, these new systems consist of individual entities or objects that jointly strive to reach a common goal, which involves sensing or the controlling of devices, and are dynamically and loosely federated for cooperation. All of this, while making sure resources are used optimally.

1.2 State of the Art in Cooperating Objects Research

In order to classify the state of the art in Cooperating Objects research, we have taken the same approach used in the Embedded WiSeNts research roadmap. In it, the relevant topics are structured in hardware, algorithms, non-functional properties and others. In this book, we have focused on research topics that, from the point of view of industrial research and the academic community, are still relevant and not considered solved.

1.2.1 Hardware

Regarding hardware, low energy processors and controllers have been designed and used, especially in the area of embedded controllers and devices. There are also advances in simple low power sensors, e.g. for temperature or humidity, while other areas are not showing similar advances, e.g. gas or air movement. However, the more efficient a particular piece of hardware is regarding energy consumption, the more expensive it becomes. Unfortunately, cost is a definite constraint in the Cooperating Object area and, depending on the application scenario, people would not be willing to pay too much for each device. Therefore, there is still a need for low-cost, power-efficient hardware.

Nowadays, the typical sensor node price lies between $50 and $200. On the other hand, applications requiring more than 100 sensor nodes dramatically increase investment costs. Silicon-based tilt sensors offer cost effective solutions over older fluid-vial sensors. The ultimate target is to produce sensor nodes with a price of under $1. Despite their small size when compared to traditional systems, they are still too big to be embedded in small objects of daily life, e.g. in smart home scenarios. System on Chip solutions with smaller dimensions exist, but are usually tailored to specific scenarios. Moreover, the key problem is often the battery and packaging and not the sensor node itself.

While some approaches for calibration have been proposed in the recent past, actual calibration solutions are often ad-hoc and require a large amount of application-specific engineering. In many cases, the calibration infrastructure is at least as complex as the sensor network itself. Significant work is needed to arrive at a systematic treatment of calibration issues. Also, ready-to-use tools are needed to support calibration in practical settings.

Current research in the field of energy harvesting tries to combine existing techniques to create more efficient power generators, although there is definitely the need to improve the energy generation capabilities of individual techniques. New materials, such as electroactive polymers, are being examined since they promise a higher energy conversion coefficient.

Finally, the issues of hardware adaptation for the optimal selection of transmit power as well as the radio channel for minimal interference are topics that are starting to gain a lot of attention in the research community.

1.2.2 Algorithms

There is a great variety of algorithms and types of algorithms that have to be revisited when dealing with Cooperating Objects technologies. In this book we have included localization, MAC, bandwidth estimation algorithms, clustering, querying and data processing, as well as algorithms for the appropriate cooperation of moving objects and robots.

Regarding localization, the state of the art shows that this field has been very prolific in the past years, providing solutions that are both range-free and range-based. Current trends try to combine individual localization techniques such as sensor nodes, RSSI, camera information, etc. into a system that provides better results as the individual parts alone. Most of the research nowadays concentrates in in-door scenarios, where most of the problems are still not solved with the appropriate level of accuracy.

Regarding Medium Access Control techniques, the literature is very vast and contains protocols that have very different goals. In general, Cooperating Objects research benefits more clearly from TDMA-based algorithms that avoid collisions by design, although this implies the existence of synchronized clocks throughout the network. The trend is towards providing efficient mechanisms to schedule the access to the medium while avoiding the latencies normally incurred by this type of protocols.

Available bandwidth estimation and monitoring is one of the essential tasks to accomplish for the development of an efficient methodology for bandwidth management, which varies with the number of nodes contending for the channel. Current trends focus on the combination of existing estimation methods with a variety of network layer protocols in order to make the resource reservation decisions more accurate.

Clustering provides an efficient and scalable network structure for collaborating sensor nodes by grouping them into a hierarchy. Such hierarchical structures are constructed by various clustering approaches at different network layers such as the MAC and the routing layer. Additionally, clustering can be combined with data processing techniques such as aggregation.

Querying is perhaps the area that has concentrated most of the interest on Wireless Sensor Network research, and as a result, a number of papers have been published on this topic. Current trends in querying look at mechanisms to efficiently distribute the query to all sensors in the network without using techniques such as flodding. For this reason, techniques based on random walks are starting to gain more interest nowadays. While previous work provides a better understanding on the performance of random walks on WSN, most of them are based on ideal communication models. Identifying a random-walk based querying mechanism that exploits that particular characteristics of WSN communication graphs is still an open area of research.

Distributed query processing is essential for prolonging the lifetime of the network. There are various in-network mechanisms that are widely used in wireless sensor network applications including aggregation, suppression and view management. More recent query processing techniques try to achieve query based routing, a technique that incorporates query semantics into its execution.

Decision making theories are relevant in mission planning, task allocation and intrusion detection. These include byzantine agents, team utility maximization, distributed negotiation protocols and optimal assignment. The basic concepts involved in sensing and perception are data fusion, optimal deployment under limited sensing for coverage, cooperative perception, rumor propagation and active perception. New control approaches of Cooperating Objects interacting with the environment are also related to environment perception and interpretation, and self-monitoring to improve reliability. All the above concepts and methods have been used in the mentioned areas, i.e. robotics, control, decision making, and communication. However, their integrated application for the coordination of mobile objects sharing the same physical space in cooperative missions is still in its infancy.

1.2.3 Non-functional Properties

Non-functional Properties (NFPs) are defined as the properties of a system that do not affect its functionality, but its quality. We consider NFPs as the Quality-of-Service (QoS) characteristics of a system, where QoS should be interpreted in a holistic way, instantiated in properties such as scalability, timeliness and real-time considerations, reliability and robustness, mobility, security and heterogeneity.

Regarding scalability, although a very large number of processors and sensors can operate in parallel and hence the processing and sensing capabilities increase linearly with the number of sensor nodes, the communication capability does unfortunately not increase linearly with the number of sensor nodes. Several research works and commercial products propose hierarchical architectural solutions for Wireless Sensor Networks. The concept of multiple-tiered network architectures has been employed since a long time ago in other networking domains. However scalability and, on a related note, large-scale deployments still remain a line of research without a clear solution.

Regarding timeliness, the general principle of real-time systems design is to ensure temporal predictability of the tasks involved in the application, and in their scheduling. Hard real-time systems require a strict worst-case execution time (WCET) analysis of the tasks (and the related worst-case transmission times for the communication aspect), while soft real-time systems can use statistical analysis based on code profiling, simulation or real experiments. A fundamental difficulty in designing Cooperating Object systems with real-time requirements results from design principles that are usually antagonist to "traditional" real-time systems. Current solutions rely on the use of contention-free MAC protocols to ensure collision-free and predictable access to the medium, and the ability to perform end-to-end resource reservation.

Algorithms used for Cooperating Objects must be reliable and robust with respect to sudden and/or long-term changes. An algorithm is robust if it continues operating correctly despite abnormalities (e.g. in inputs, calculations). Algorithms used for routing, localization, mobility, etc., should keep working properly even if operational conditions or the structure of the system change. Most fault avoidance techniques operate in the network layer, adding redundancy in routing paths; a majority of fault detection and recovery techniques operate at the transport layer; and a few fault recovery techniques perform at the application layer, concealing faults during off-line data processing. In order to provide a higher level solution for fault-tolerance, fault-management frameworks with complete management infrastructures and information models have been currently proposed and will continue to be studied in the next years.

Physical mobility mainly refers to the changes of the geographical locations of an entity during time. Logical mobility refers to the dynamic changes in the network topology such as adding or removing new entities in the system. There are three types of mobility: node mobility, sink mobility or event mobility. Generally speaking, many routing algorithms are able to cope with topology dynamics resulting from nodes mobility. However, most of them react to topology variations by dropping the broken paths and computing new ones from scratch, thus incurring in performance degradation. In particular, mobility may strongly affect cluster-based algorithms, due to the high cost of maintaining the cluster-architecture over a set of mobile nodes. Some routing algorithms specifically designed for networks with slow mobile nodes (e.g. GAF, TTDD) attempt to estimate the nodes trajectories.

Given the interactive and pervasive nature of Cooperating Objects, security is one of the key points for their acceptance outside the research community. Security in Cooperating Objects is a more difficult long-term problem than is today in desktop and enterprise computing. In the normal case, there is no central, trusted authority that mediates interaction among nodes. Furthermore, Cooperating Objects often use wireless communication in order to simplify deployment and increase reconfigurability. So, unlike a traditional network, an adversary with a simple radio receiver/transmitter can easily eavesdrop as well as inject/modify packets in a wireless network. Current research topics in the area of security include the problem of bootstrapping security, key distribution and revocation, secure configuration of devices, efficient intrusion detection and secure routing. Most of the solutions available currently only provide partial solutions tailored to specific systems.

When speaking of heterogeneity, the literature considers it at different levels: heterogeneous networking hardware and software, heterogeneous embedded devices, heterogeneous infrastructure and heterogeneous applications / services. Current solutions are developed taking into account the specific task at hand and do not attempt to provide generic approaches.

1.2.4 Systems

Under systems we consider three types of software: operating systems, middleware, and system integration tools, including diagnosis and debugging mechanisms. In general, these areas have received a lot of attention in the past years and have been very prolific.

Regarding operating systems, the trend is towards the creation of more and more complex system software that is able to deal with the resource limitations of Cooperating Objects while at the same time offering a wide range of functionality (even threading and real-time scheduling). The main constraints are at the device level where operating systems like TinyOS or Contiki have to be used as opposed to bigger systems (such as robots) where embedded Linux variants are feasible.

The second type of system deals with the development of middleware solutions that extend the capabilities of the operating system by offering certain services and abstractions that can be used by a wide variety of applications. There are numerous types of middleware systems that can be classified in: macroprogramming mechanisms, virtual machines, network level abstractions, task distribution solutions, adaptive systems, data sharing abstractions, service invocation, event detection and context management. Additionally, there are middleware solutions that mediate and act as system integration solutions for heterogeneous devices and networks. The trend towards integrative solutions is in line with the increase in heterogeneity seen in current deployments.

Regarding debugging and inspection tools, there are three different types of solutions: active inspection, passive inspection and self-inspection solutions. The field of non-intrusive debugging is receiving a lot of attention in the past years and has been the major topic of important conferences in the areas of Wireless Sensor Networks.

Finally, some systems are able to heal themselves upon detection of a problem. Healing can be performed at the protocol level, by updating the software executing on the nodes, or at the physical level, for example, by relocating nodes.

1.2.5 Others

Other topics relevant from the point of view of research are modeling and planning of static and mobile networks and topologies, as well as testbed and simulation platforms and standardization practices. All of them form encompassing solutions and tools that, although crucial for the use and development of Cooperating Objects technologies, do not fall under the previously mentioned categories.

Regarding planning, there are a series of solutions that deal with the pre-deployment of networks by using either analytical methods, simulation tools or small testbed deployments. Some of these solutions attempt to perform the planning of the network lifetime instead of planning the position of nodes based on communication capabilities, sensing ranges, etc.

Simulation and testbeds are indispensable tools to support the development and testing of Cooperating Objects. Simulations are commonly used for rapid prototyping which is otherwise very difficult due the restricted interaction possibilities with this type of embedded systems. Simulators are also used for the evaluation of new network protocols and algorithms. Simulations enable repeatability because they are independent of the physical world and its impact on the objects. Simulations also enable non-intrusive debugging at the desired level of detail. However, it has been shown that the models used for mobility, traffic, and radio propagation have a significant impact on the simulation results.

There are three types of simulators that can be used for the development of Cooperating Objects technologies: generic simulators such as ns-2, specialized simulators that deal with a specific part of the technology such as MAC protocols, hardware platforms, etc., and emulators of hardware devices. The type of simulator/emulator that should be used depends on the task at hand. Current trends deal with the combination and integration of simulators based on their individual characteristics in order to create better and more effective simulation results.

The primary goal of a Cooperating Objects testbed is to support the design, implementation, testing and evaluation of applications and protocols without forcing the investigator to make artificial assumptions about system components or the system environment (as often needed in analytical and simulation work). A successful testbed architecture needs to accommodate the specifics of Cooperating Objects in a scalable and cost-efficient way. Currently there are several dozens of testbeds deployed world-wide with different levels of software abstractions, capabilities, etc., and these numbers are increasing rapidly.

Finally, regarding standards for Cooperating Objects, there is currently a series of international associations that deal with standardization efforts for small devices. Organizations such as the ZigBee Alliance, the IPSO Alliance of ETSI are contributing very actively to the standardization of relevant software ranging from communication protocols, to network protocols or even web service descriptions.

1.2.6 Conclusion

As can be seen, Cooperating Objects research puts together a series of highly dynamic and multi-disciplinary areas that cover aspects of both hardware and software, as well as their integration into functioning systems that work in the real world. Given the relative youth of the field, there is also need to perform research on the supporting tools that enable the programming, debugging and integration of such systems.

Although we have tried to cover as many aspects of the field as possible given the expertise of the authors, it seems clear that this overview of state of the art cannot contain all aspects of research. Nevertheless, we are confident that we have been able to select some of the most promising ones from the point of view of industry and academia.

1.3 Innovative Applications

The CONET consortium also has looked at the most promising applications whose innovation factor can be followed back to Cooperating Object technologies. The level of detail as well as the degree of visionary foresight depends on the amount of work industry has put into the different application areas. Therefore, some of them are more mature than others.

Industrial Building and Automation: This group of applications encompasses not only building automation but also home control and industrial automation. The innovation in these applications lies in the integration and combination of mesh networks and Cooperating Object technologies in order to create hierarchical systems that can be used in large-scale deployments.

Energy: These applications deal with the optimal use of power by the use of smart meters or collections of meters that collaborate with each other in order to implement a Cooperating Object network. The innovation in these applications is tied to the creation of a distributed infrastructure for the production and consumption of energy that can be channeled optimally to the right users while taking into account average as well as peak consumption.

Transportation: This group of applications deal not only with traffic scenarios but also look at aerial transportation and the control problems associated with it. Their main innovation factor have to do with system wide and large-scale traffic information systems as well as the deregulated and completely distributed control of vehicles or airplanes with guarantees.

Environmental Monitoring: These applications are among the more classic ones for Wireless Sensor Network technologies and have been a part of active deployments since their conception in the 90s. They have been classified as: large-scale single function networks, localized multi-function networks, bio-sensor networks and heterogeneous networks. The main innovation factor of these applications have to do with the combination of large deployments and the need for heterogeneous hardware that works in a robust way over long periods of time.

Healthcare and assisted living: This group of applications encompasses scenarios that deal with the improvement of assisted living techniques, activity and/or emotion recognition and gait analysis. The main innovation factors in this area have to do with the correct analysis and prediction of complex behavior that can be used to improve the life of people in their own environments. This should be performed in a non-intrusive way and, at the same time, in a distributed way since not a single entity will be able to gather all available data.

Security: This group of applications include the detection, identification and classification of targets, as well as the tracking of possible intruders. The innovation in this area comes from the distributed processing of information and from the robustness and reliability required for the proper detection of intruders in an area monitored by Cooperating Objects.

It seems clear from the wide range of applications presented that the reach of Cooperating Objects technologies extends to almost every aspect of our daily lives. However, these are just the tip of the iceberg if Cooperating Objects manage to become one of the mainstream technologies in the next years. In general, all applications that can increase performance of functionality from the cooperation of smart embedded devices, will benefit from the line of research followed by the Cooperating Object community. In order for this to happen, the market has to be ripe for the technology and the appropriate research gaps will have to be closed to a level that can be used by the industry in order to create products and applications that can be installed in the real world.

1.4 Market Analysis

According to ON World Inc., the global market for Wireless Sensor Network systems and services is expected to skyrocket to about $4.6 Bn in 2011, up from approximately $500 million in 2005. There will be a worldwide (conservative estimate) market of $5.3 Bn for the industrial control segment only, comprising 4.1 Million nodes by 2010. ON World Inc. most aggressive forecast for all wireless sensor (& control) network segments is $8.2 Bn by 2010, comprising 184 Million deployed nodes. It is important to note that ON World Inc. projections only account for the physical node hardware shipments - not the physical gateway hardware, nor any independent system software components, enterprise software components, system integration services or other ancillary services.

Furthermore, the survey performed by the CONET consortium confirms the potential of Cooperating Objects as well as providing an insight as to which roadblocks will need to be solved in order for Cooperating Objects technology can become mainstream. The most important ones are the lack of clear business models, lack of standards, as well as lack of confidence in the technology, which leads to unresolved social issues.

The majority of the market growth predictions were made before the economic meltdown of the late 2008 and 2009. As such, the aforementioned numbers should be taken as an indicative trend in the market and show its potential; the future will tell if and at what timeline they will be validated.

Nevertheless, it is clear that there is promising potential in versatile domains, that could greatly benefit with the introduction of Cooperating Object technologies, ranging from automation (home, industrial, building) to healthcare, energy etc. We expect that the Cooperating Objects market will be cross-domain and strongly embedded in the fabric of success of other domains.

1.5 Research Roadmap

Using as a basis the analysis of the state of the art, the review of innovative applications and the market analysis we have been able to identify the predominant areas that will need attention in the next years. Additionally, we discuss the results of our own estimation and the one from a series of surveys conducted among experts that indicate the approximate time where these gaps are expected to be solved.

1.5.1 Gaps and Trends

The following gaps have been identified and classified using the categories presented in the previous sections:

- Hardware:
  - Development of energy efficient hardware
  - Energy harvesting techniques
  - Miniaturization of hardware
  - Adaptation to resources of hardware
- Algorithms:
  - Integration of UWB into Cooperating Objects
  - Cost-effective localization mechanisms
  - Data processing techniques for large and heterogeneous networks
  - Support for multiple data sinks
  - Motion planning for resource constrained devices
  - Resource adaptation for MAC protocols
  - Efficient and distributed bandwidth estimation techniques
- Non-functional Properties:
  - Scalability: Efficient MAC, routing and data processing algorithms for large-scale deployments
  - Timeliness: Real-time features for Cooperating Objects
  - Reliability / robustness: Fault-tolerant mechanisms that spread across different layers; recovery mechanisms for channel and node failures; proactive adaptation of routing paths
  - Mobility: Time and energy-efficient mobility support;
  - Security: Light remote program integration verification

  - Heterogeneity: Support of heterogeneity across all levels of hardware and software layers

- System:
  - Operating systems available and suitable for all sizes of Cooperating Objects
  - Support for real-time operation
  - Self-optimization, self-monitoring and self-healing approaches
  - Formal verification of the correctness of operating systems
  - Mechanisms to combine different middleware solutions
  - Development of a Cooperating Objects software "construction kit"
  - Adaptive systems with cross-layer support
  - Common functionalities and interfaces for the integration of systems in real deployments

- Other:
  - Deployment of nodes and their continuous monitoring
  - Better algorithms for the estimation of the lifetime of deployments
  - Integration of diagnosis and healing mechanisms
  - Better integration of diagnosis with programming tools
  - Accurate mobility models for simulation / emulation
  - Open implementation missing from many standards

1.5.2 Timeline

The estimated timeline for the solution of these gaps was derived from our own experience in the field and from the surveys performed among independent experts at different events.

Regarding hardware, we expect Sensor Calibration to be solved relatively soon in comparison to other gaps because unless this issue is solved in a satisfactory way, it is hard that sensors can be used in environments where costs play a major role, such as in the Home and Office domain. Other more industrial domains are willing to pay higher prices and, therefore, more sophisticated methods for sensor calibration can be used. All other gaps are expected to be solved in the mid-term or long-term.

For the algorithms area, localization and MAC and routing have received lots of attention in the past years, so the expectancy is for these problems to be solved in the short-term. Other types of algorithms, like motion planning or role assignment will take longer to solve.

Research on non-functional properties such as improving the timeliness, security and reliability/robustness of Cooperating Object systems are still at a very early stage, particularly for the latter. Scalability is being considered by researchers (e.g. algorithms, methodologies, protocols), but results are still either incomplete, immature and/or yet to be validated in real-world applications. Almost no work exists on supporting mobility (nodes, node clusters) in Cooperating Object systems. While successful results are not obtained using homogeneous Cooperating Object systems, it will be hard (almost impossible) to support high levels of heterogeneity, such as the coexistence and interoperability between different hardware platforms, network protocols, operating systems, middleware and applications.

As for systems, Operating Systems will be solved soon since they are the basis for all Cooperating Objects software. On the other hand, middleware solutions, programming models and adaptive systems will be relevant in the medium and long term. The same holds for diagnosis and healing capabilities of these networks.

Finally, except for simulators and emulators which are expected to reach consensus in the short run, and standardization processes that will require a long time to converge, all other gaps in terms of modeling and deployment capabilities of Cooperating Objects are considered to be solved in the medium to long term.

1.6 Predominant Work Areas

Although the gaps identified in the previous chapter still hold and need to be solved, the topics shown in this chapter should receive the most attention in the following years in order to advance the area of Cooperating Objects in the most effective way.

- Energy considerations:
  - Research on battery lifetime and energy storage
  - Energy-aware and power-efficient hardware
  - Power efficient algorithms
- Localization:
  - Accurate in-door localization mechanisms
  - Seamless transition of location information between in-door and out-door deployments
- Data management:
  - Handling of large amounts of data
  - Cross-layer optimizations for data processing

  - Optimization for the planning, routing, processing and storage of queries and data

- Non-functional properties:
  - Application-specific classification and optimization of non-functional properties like timeliness and robustness
  - Security aspects for resource-constrained devices
- System support:
  - Support for adaptation at the operating system and system software level
  - Diagnostic and debugging approaches that can be used continuously, if necessary
- Modeling and planning:
  - Planning tools for the deployment of sensor networks with guarantees
  - Realistic modeling tools
- Simulators and testbeds:
  - Integrated simulators that allow for a combination and comparison of test results in an easy way
  - Integration of testbed and their capabilities for the interchange of code and tests
- Standardization:
  - Consolidation of current standards and their widespread adoption

In all domains of Cooperating Objects research areas have been identified that need to be reinforced since their solution is vital for the adoption of Cooperating Objects. Many proposed predominant work areas do not only cover a single topic but present different and interdependent domains. Strong collaboration between different researchers in different domains is, therefore, necessary to tackle these complex tasks.

1.7 Purpose and Intended Audience

The document you have in your hands presents the vision of the CONET consortium and its associated industrial partners regarding the future development of research in the field of Cooperating Objects. This vision is presented in the form of a technology roadmap and is the result of the compilation of several factors:

- The individual **expertise** and **practical experiences** of each of the partners involved in the project;

- the analysis of the current technologies and **current trends** that show future research directions.
- A **market analysis** of Cooperating Objects performed with the input from industrial partners and other research institutes.
- **Innovative applications** obtained partially from a wider audience and from within the consortium.
- **Identification of gaps** and research agendas in the different areas that compose the field of Cooperating Objects.

Given the balanced research and industrial background of the contributors and the fact that the field of Cooperating Objects is advancing rapidly, this document should be seen as input for research and development departments in industry and academia that would like to benefit from information about the possible direction and the timeframe for Cooperating Objects research.

The CONET Research roadmap has been written with three different audiences in mind:

- Researchers: That work or intend to work in the field of Cooperating Objectsand would like to understand the current state of the art, current trends and possible gaps for future research.
- Industry: That would like to understand the current state of the art and possible market developments to be used as an additional source of information for the definition of specific strategies and business opportunities related to Cooperating Objects.
- R&D Managers and policy directors: To achieve a better understanding of the field of Cooperating Objects and its potential as a topic that can be included in upcoming calls or other financing instruments.

Depending on the interest of the reader and its adhesion to one or more categories described above, the reader should select the chapters and sections that most fit his/her interests.

Chapter 2

Introduction to Cooperating Objects

A number of different system concepts have become apparent in the broader context of embedded systems over the past couple of years. First, there is the classic concept of **embedded systems** as mainly a control system for some physical process (machinery, automobiles, etc.). More recently, the notion of pervasive and **ubiquitous computing** started to evolve, where objects of everyday use can be endowed with some form of computational capacity, and perhaps with some simple sensing and communication facilities. However, most recently, the idea of **Wireless Sensor Networks** has appeared, where entities that sense their environment not only operate individually, but collaborate together using ad hoc network technologies to achieve a well-defined purpose of supervision of some area, some particular process, etc.

We claim that these three types of systems (i.e. embedded systems, pervasive and ubiquitous computing and wireless sensor networks) that act and react on their environment are actually quite diverse, novel systems that, on the one hand, share some principal commonalities and, on the other hand, have some different aspects that complement each other to form a coherent group of objects that cooperate with each other to interact with their environment. In particular, important notions such as control, heterogeneity, wireless communication, dynamics/ad-hoc nature, and cost are present to various degrees in each of these types of systems.

The conception of a future-proof system would have to combine the strong points of all three system concepts at least in the following functional aspects:

- Support the control of physical processes in a similar way embedded systems are able to do today.

- Have as good support for device heterogeneity and spontaneity of usage as pervasive and ubiquitous computing approaches have today.

- Be as cost efficient and versatile in terms of the use of wireless technology as Wireless Sensor Networks are.

The convergence of these three types of technologies that, until now, have been evolving independently of each other (Figure 2.1), is what we call Cooperating Objects technologies. This new term is born out of the combination of these traditional systems.

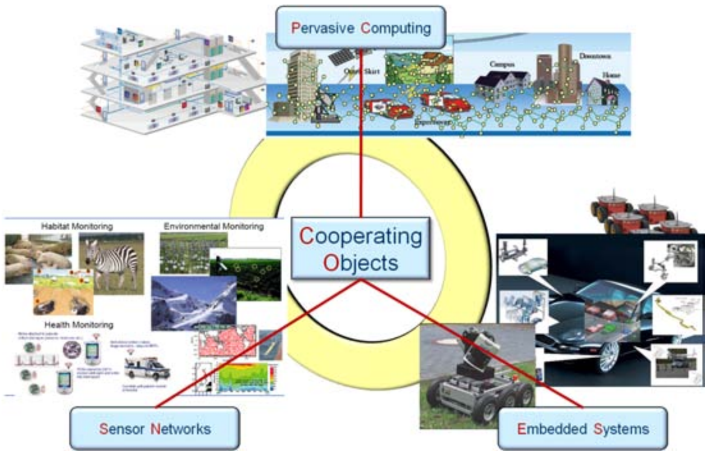

Figure 2.1: *Areas covered by Cooperating Objects*

Moreover, this notion or paradigm of Cooperating Objects is even stronger than the individual technologies it stems from, as it carries over to their internal structure – e.g. a Wireless Sensor Network can be regarded as consisting of Cooperating Objects itself, highlighting the diversity of cooperating patterns admissible under this general paradigm. Also, pointing to the importance of complementing the vision of pervasive computing with that of pervasive control is essential.

2.1 Definition

Following the concepts we have just discussed, let us now define more formally what a Cooperating Object is. In the abstract sense, a Cooperating Object (CO) is a single entity or a collection of entities consisting of:

- **sensors**, devices that act as inputs to the Cooperating Object and are able to gather and retrieve information either from other Cooperating Objects or from the environment;

- **actuators**, devices that act as output producers and are able to interact and modify their environment;

- **controllers** (information processors), devices that act as data or information processors and, obviously, must interact with sensors and actuators in order to be able to interact with their environment;

- or **Cooperating Objects**.

All entities communicate with each other and are able to achieve, more or less autonomously, a common goal.

Furthermore, controllers are equipped with some kind of storage device that allows them to perform their tasks. The amount of effort devoted by a particular controller to either information processing or storage tasks is determined on an individual basis. This is the main difference with respect to other related technologies such as RFID. In the case of Cooperating Objects, the intelligence of the system lies distributed in the network and each individual entity is able by design to perform complex processing tasks, if so needed. On the other hand, RFID does not perform any kind of processing and only returns an identification as an answer to an external stimulus (the reader). In this sense, the intelligence of a system based on RFID technologies lies in the infrastructure and in the readers, but not on the distributed and embedded devices that form the bulk of the network.

It seems clear that if sensors, controllers and actuators need to interact with each other in a distributed environment, all of them need to be equipped with communication capabilities. These might of course be based on wired or wireless technology.

The inclusion of other Cooperating Objects as part of Cooperating Object itself indicates that these objects can combine their sensors, controllers and actuators in a hierarchical way and are, therefore, able to create arbitrarily complex structures.

In view of the emergence of new technologies and devices, their increasing integration into the everyday life and the need to coordinate them with a view to making communication easier mainly as to the interoperability, the mobility and the scalability, Cooperating Objects are regarded as a key enabler and aim at providing a proactive support to users or machines in their collaborative tasks. Indeed, the major advantage of the Cooperating Object lies in the possibility to tackle the complexity of the new surrounding environments due to the high number of involved devices or systems and the heterogeneity of components.

The generality of this model allows us to seamlessly include different fields like sensor networks, pervasive computing, embedded systems, etc. However, depending on the way we look at the algorithms and systems developed for Cooperating Objects, we can define **data-centric** and **service-centric** approaches.

As an example, consider the following scenario: Nowadays, we have at or disposal lots of information, data sources or systems or even services, like the traffic panels along the main roads, traffic radio, GPS devices or web services to plan a trip. GPS devices may give alternatives but only under the human initiative and not all commercial GPS take into account real-time traffic data yet. However, video cameras and other surveillance systems are able to provide some of this data. And if we integrate the Traffic Message Channel services (TMC technology) for example, we get another useful flow of information which could be computed. To be able to achieve our goal, a pro-active process is necessary, and for this reason the co-operation between all of these various information sources is vital. How to reach that, i.e. a cooperative surrounding environment to link vehicle and infrastructures? One of the solutions would be to use Cooperating Objects, which would make it possible to drop some current barriers between these elements such as heterogeneity, complexity, scalability and to improve the communication with a view to providing ad hoc networks, thus data mobility would be enhanced. In this scenario, we could have various Cooperating Objects: for instance one that continuously measures local traffic data, a second one to integrate all traffic-related data from available information flows from infrastructures and another that asks the GPS device in view to offering route alternatives. The first one is an example of a classical sensor network, whereas others would be traditionally described respectively as a controller network and an actuator.

2.2 Data-centric Approaches

The field of Wireless Sensor Network research is the canonical example of data-centric approaches. In this field, the efficient management of data is in the core of all published algorithms. Additionally, some other characteristics are relevant for sensor networks, such as:

- **Minimal user interaction:** Given a query from the user, a sensor network should be able to autonomously and automatically figure out the most efficient way to provide an answer to the user.
- **Resource-limitation:** Sensors have usually limited resources in terms of energy, capacity, sensing capabilities, etc.
- **Ad-hoc organization:** Most sensor networks expect their nodes to be able to communicate with each other without the use of any kind of infrastructure.
- **Wireless communication:** As a consequence of the ad-hoc nature of sensor networks, communication is usually performed using wireless technology.

In general, data-centric approaches are chosen in environments where the naming of data and the use of data types within the network play a more important role than the specific node that might be responsible for its processing. Therefore, there is a decoupling of data and network node that can be used to dynamically select the appropriate location where data processing is performed without it affecting the expectations from the user. Data-centric approaches are best suited for database-like operations such as aggregation and data dissemination.

In the literature, there are two different kinds of data-centric processing techniques. The first one uses the query/response (or request/reply) paradigm, so that the network of Cooperating Objects only sends responses to specific queries issued by the user. In the second technique, queries are stored in the network and are provided with an associated lifetime.

During their lifetime, each sensor is responsible for the processing of the stored (or continuous) query and sends messages to the issuer of the query (also called sink) whenever the condition specified in the query is met. Therefore, both pull-based and push-based approaches can be used in data-centric environments.

Although the absolute position of nodes within the network do not play an important role from the perspective of the user (or the issuer of the query), good topology management techniques need to be used in order to maximize the lifetime of individual sensors. It is crucial to know where the neighbors are and what kinds of roles they play in the network in order to optimize the processing of queries.

In the context of CONET, it is necessary to study this class of approaches in order to examine their degree of applicability and their potential benefits related to the problems that these approaches seem to alleviate in the framework of embedded systems or sensor networks, such as, for instance, scalability (regarding data acquisition or data aggregation), distributed aspect of systems and also fault tolerance.

2.3 Service-centric Approaches

In contrast, service-centric approaches are mostly concerned with the definition of the interface or API in order to provide functionality for the user. Depending on the specific fields there are other additional characteristics that need to be mentioned. For example, in the field of pervasive computing, the miniaturization of devices as well as resource limitation play an important role, whereas in classic client-server architectures no such restrictions apply.

In such environments, the transport mechanisms are hidden from the user applications (such as in traditional networked environments), but a certain cooperation among the nodes in the network allows for the processing of data. The difference to data-centric approaches lies in the kind of programming techniques needed to interact with the network. In a service centric environment, the application developer is supposed to have and use a clear specification of services offered by the network.

Also in these types of environments is the use of pull-based and push-based approaches widespread. The use of traditional APIs would cover the case of pull-based interactions, whereas a publish/subscribe mechanism would provide the necessary APIs needed for a system to interact with the user in a push-based fashion. The specific location of a service needs to know where services are located, or needs to be able to contact a location service that knows where services are located.

Finally, the implementation of real-time APIs and real-time constraints has been studied in much more detail since the enforcement of such constraints can be performed through API implementations.

In CONET we consider to deal with those approaches, most likely in the framework of broadband services, including wireless technologies like Wi-Fi, WISPs, or also 4G technologies, or context-aware services helping the monitoring activity and personalization. Our goal aims to explore the benefits and the opportunities that they may bring, regarding for example, real-time, interoperability, mobility, sustainability or scalability.

2.4 Enabling Technologies

Nowadays, it is impossible to create or work on technologies that do not rely more or less heavily on the development of other areas. New developments in these related areas usually go hand-in-hand, and a major breakthrough in one of the enabling technologies can really boost the work that can be performed on the other areas.

This is also true for Cooperating Objects and, as we have seen in the previous sections, Cooperating Objects have emerged as a combination and natural extension of already existing research areas that have been evolving rapidly in the past years.

Therefore, it is worth pointing out more precisely what we consider are the major pillars for research in Cooperating Objects, so that the readers can keep them in mind while reading the following sections.

Figure 2.2 shows the four pillars that, in our view, support work on Cooperating Objects. These are: miniaturization, power sources, communication and smartness. Let us now describe in more detail what the purpose and relationship of each pillar is with respect to Cooperating Object technology.

Miniaturization: Research on miniaturization deals with the creation of always smaller sensors, actuators and, in general, devices that can be used to implement a network of Cooperating Objects. If the long-term vision of Mark Weiser of the disappearing computer is to become true, the miniaturization of devices plays a crucial role by implementing more into less space. The newest developments in this area even discuss now the possibility of incorporating a whole network of devices into a single chip (nets-on-chip), making them ideal candidates for their incorporation in Cooperating Object research. Miniaturization is definitively an enabling technology for systems such as robot swarms, RFID supporting

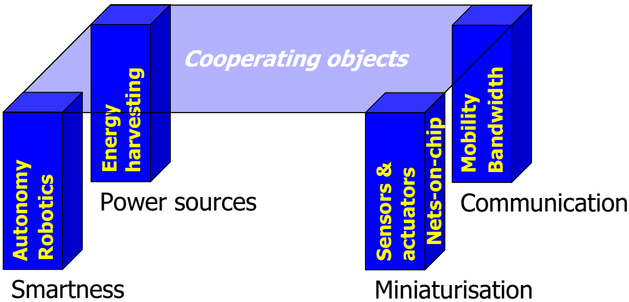

Figure 2.2: *Enabling Technologies for Cooperating Objects*

deployment or Emotion-aware Ambient Intelligence (AmI) using microsensors or implants in the field of Pervasive and Ubiquitous Computing.

Power Sources: Related to the previous enabling technology, research on power sources seems to be one of the major concerns when designing smaller and smaller devices. Current research on batteries and energy sources cannot really keep up with microcontroller technology and, when discussing this issue with hardware experts, they complain that hardware could be much smaller if they just had a way to power it properly. Following this suggestion, a considerable amount of effort is being put on energy harvesting techniques that use vibrations, electro-magnetic waves, motion, etc. to power small devices. Battery technology is a very active research field as well. Also, significant effort is dedicated to the development of energy efficient wireless communication protocols. The good news is that, in most cases, the smaller the devices, the lower the amount of energy is that needs to power it, but current research has not yet found the sweet spot where miniaturization and battery technology can be scaled down together to the sizes needed for the implementation of Cooperating Object technology.

Communication: Research on communication technologies has received a lot of attention in the past decade. The research community has produced highly efficient algorithms for the transmission of data between computers. However, the characteristics of Cooperating Objects, especially the fact that devices have to communicate with each other in order to be able to do anything interesting, and the sheer amount of devices that need to communicate, has changed the characteristics and metrics that make communication algorithms efficient. New Quality of Service (QoS) metrics have been identified, such as energy consumption, bandwidth or mobility, that make this field not only an enabling technology for Cooperating Objects, since communication is crucial for its operation, but also have also created new research directions that can be followed independently of explicit research on Cooperating Object technology. Nowadays, communication is one of the key drivers to create, develop or use technologies. Indeed, numerous emerging technologies could have impact on mobility, deployment, adaptation and personalization.

Smartness: Also related to communication and QoS is the fact that cooperation needs to happen in an unknown (sometimes even hostile) environment. Therefore, smartness is an enabling technology for Cooperating Objects since communication is definitely necessary but the need for smart behavior and efficient cooperation is definitely needed from the Cooperating Objects that make up a network. In this area, the autonomy achieved by current robot technology, or the fact that the system and each device needs to sense and adapt itself to its environment, make a certain degree of smartness a required characteristics and, therefore, an enabling technology, for Cooperating Objects.

Smartness is also another key driver for the innovation. For this pillar, we can quote for instance, smart actuators, autonomous mobile robots and also the MEMS technology regarded as an enabling technology allowing the development of smart products.

Chapter 3

State of the Art in Cooperating Object Research

This chapter provides an overview of the State of the Art in Cooperating Object research and, thus, serves as basis for subsequent examinations of the research gaps. Although it tries to give a broad overview of Cooperating Object research it does not cover aspects that seem to be solved from the point of view of academic or industrial research. Having this in mind, we present the State of the Art in hardware, algorithms, non-functional properties, systems and other aspects of Cooperating Objects.

3.1 Hardware

A Wireless Sensor Network hardware node is commonly referred to as a Mote. The dictionary definition of the word Mote is a very small particle; a spec, which describes well the aspirations for Wireless Sensor Networks for Cooperating Objects. The architecture of a typical wireless sensor node is depicted in Figure 3.1. The node is comprised of four key components: power supply component, a sensing component, a computing component and a communication component. In this section we consider several aspects related to these components.

3.1.1 Sensor Calibration

3.1.1.1 Description and Relevance

Low cost is a key requirement for the widespread adoption of sensor networks for real-world applications. However it is clearly important that the output of a sensor network mirrors the ground truth of the real world, and this is what sensor calibration tries to address.

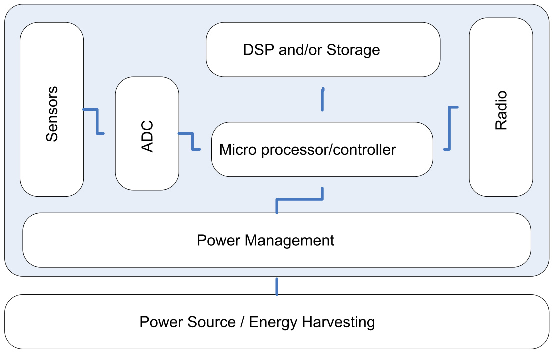

Figure 3.1: *Typical architecture of a wireless sensor node*

There are two issues to consider here. First, it must be ensured that sensor output is consistent across the network, i.e., that two different sensors output the same value if they are presented the same physical stimulus, this is called relative calibration. Secondly, we have to make sure that sensor output matches the real world, i.e., if the ambient temperature is 25 ℃, then a temperature sensor should also report the equivalent of 25 ℃, this is called absolute calibration.

Relative calibration can exploit the fact that some physical quantities change only slowly over distance (e.g., temperature), such that sensors which are close together should report similar values. Absolute calibration is typically achieved by temporarily installing a second, calibrated measurement device alongside the sensor network to measure ground truth. For example, many installations use a set of video cameras to observe what is happening in the network and to calibrate the output of the actual sensor network. Additionally, other settings use mobile nodes with appropriate sensors to perform the calibration of static nodes.

3.1.1.2 Existing Trends

While some approaches for calibration have been proposed in the recent past, actual calibration solutions are often ad-hoc and require a large amount of application-specific engineering. In many cases, the calibration infrastructure is at least as complex as the sensor network itself. Significant work is needed to arrive at a systematic treatment of calibration issues. Also, ready-to-use tools are needed to support calibration in practical settings.

3.1.2 Power Efficiency

3.1.2.1 Description and Relevance

Power has been one of the most important issues for the electronic world since the very beginning. The modern world has almost no problem with supplying power for fixed systems, like servers or home PCs, but mobile systems are a different issue. Usually, batteries are used that need to be periodically recharged. Since this is infeasible in some scenarios, many new technologies have been created to feed these systems with power (see Section 3.1.3).

Power efficiency should, therefore, be examined from multiple different points of view including power efficient hardware, software and system design. For example the power consumption of the receive portion of the radio is typically a significant contributor to the overall power budget, and for a high percentage of the time this energy is wasted. Current battery technology is able to store enough power for long hours of discontinued operation, but their capacity is limited by their size. Unfortunately, battery size has to be properly scaled to the size of the device it is used on and, therefore, smaller devices do not have the batteries yet that would allow them to achieve the lifetime needed for their continuous operation. On the other hand, power efficient algorithms are needed to be able to make appropriate use of the available power and not to drain the battery without a real need.

Cooperating Object systems require the cooperation of several (possibly mobile) devices that work together to perform a common task. However, the individual components of such systems are very small, and so are their batteries. Connecting them to a wired power supply is not possible and the use of alternative power sources such as the sun (solar cells) is not always a solution. Additionally, these devices are required to adapt to their environment and not to rely on specific power sources such as solar energy.

Moreover, Cooperating Objects communicate with each other using wireless technology and, depending on the deployment, even communicate continuously over extended periods of time. Therefore, the need for power-efficiency hardware devices and power-efficient algorithms are two of the most important gaps related to Cooperating Object research.

3.1.2.2 Existing Trends

Regarding hardware, low energy processors and controllers have been designed and used, especially in the area of embedded controllers and devices. There are also advances in simple low power sensors, e.g. for temperature or humidity, while other areas are not showing similar advances, e.g. gas or air movement. However, the more efficient a particular piece of hardware is regarding energy consumption, the more expensive it becomes. Unfortunately, cost is a definite constraint in the Cooperating Object area and, depending on the application scenario, people would not be willing to pay too much for each device. Therefore, there is a need for low-cost, power-efficient hardware. Regarding power-efficient software and algorithms, the research community is currently working on individual solutions for specific problems. For example, an application might be able to switch off certain parts of the hardware if it knows it will not need it in the next time. Using this technology, the radio interface could be switched off if it is not used for an extended period of time. In other approaches, data is aggregated within a cluster and then sent to the cluster head that takes care of sending it to the appropriate recipient. However, most of these solutions only work in certain conditions, for specific types of algorithms, etc. and lack a generic solution that can be used in a wide variety of application domains.

3.1.3 Energy Harvesting

3.1.3.1 Description and Relevance

The powering of remote and wireless sensors is widely cited as a critical barrier limiting the uptake of Cooperating Objects. Replacing batteries can become a major time consuming task that can also be uneconomical and unmanageable. There are also many applications where battery changes are not practical, such as biomedical implants and structure-embedded micro-sensors (e.g. corrosion sensors embedded in concrete or strain sensors placed on an airframe).

Energy harvesting is a means of powering electronic devices by scavenging many low grade ambient energy sources such as environmental vibrations, human power, thermal and solar and their conversion into usable electrical energy. Energy harvesting devices are therefore potentially attractive as replacements for primary batteries in low power wireless sensor nodes.

Energy harvesting technologies currently available or under development include mechanical (electromagnetic, piezoelectric and electrostatic), light (indoor and solar), thermal, electromagnetic flux and human powered. Each only suits certain application scenarios and some have yet to produce useful amounts of energy for practical application.

An energy harvester generally comprises three main components: the micro-generator which converts ambient environment energy into electrical energy, the voltage booster which raises and regulates the generated voltage, and the storage element which can be a super-capacitor or a battery.

Mechanical Energy Mechanical energy harvesting devices produce electricity from vibration, mechanical stress and strain of the surface the sensor is deployed on. Energy extraction from vibrations is typically based on the movement of a "spring mounted" mass relative to its support frame. Mechanical acceleration is produced by vibrations that in turn cause the mass component to move and oscillate (kinetic energy). This energy can be converted into electrical energy via a magnetic field (electromagnetic), strain on a piezoelectric material or an electric field (electrostatic). Most vibration-powered systems rely on resonance to work, which implies a peak frequency at which the system derives most of its energy.

Light Energy Ambient light can be used by photovoltaic cells to produce electricity either indoors or outdoors. Photovoltaic cells are exploited across a wide range of size scales an power levels. The challenge is to conform to small surface area and its power output strongly depends on environmental conditions i.e. varying light intensity. In photovoltaic cells the energy of the absorbed light is transferred to the semiconductor where it knocks electrons loose, allowing them to flow freely. The Networked and Embedded Systems Lab (NESL) at the University of California for example has developed solar harvesting hardware called Heliomote for the Mica motes.

Thermal Energy Thermoelectric energy harvesters exploit the Seebeck effect, according to which electricity is generated from a temperature difference between opposite segments of a conducting material. Temperature differentials result in heat flow and, consequently, charge flow, since mobile, high-energy carriers diffuse from high- to low- concentration regions. Thermopiles consisting of n- and p-type materials electrically joined at the high-temperature junction are therefore constructed, allowing heat flow to carry the dominant charge carriers of each material to the low temperature end, establishing in the process a voltage difference across the base electrodes. The generated power and voltage are proportional to the temperature differential and the Seebeck coefficient of the thermoelectric materials.

Electromagnetic Energy In cities and very populated areas there is a large number of potential RF sources: broadcast radio and TV, mobile telephony, wireless networks, etc. It is possible to collect parts of these disparate sources and convert them into useful energy. The conversion is based upon a special type of rectifying antenna, known as a rectenna that is used to directly convert microwave energy into DC electricity.

In laboratory environments, efficiencies above 90% have been observed with regularity. However, the energy levels actually present are so low that no present electronic device can use them.

Energy from the Human Body The human body continuously moves and radiates heat. Even at rest the human body is emitting about 100W into the environment. It is possible to tap into some of this energy to power wearable electronics. One may distinguish between active and passive energy harvesting methods. The active powering takes place when the user of the electronic product is required to perform a specific task or work they would not normally have carried out. The passive powering of electronic devices harvests energy from the users everyday actions e.g. walking, breathing, body heat, blood pressure and finger motion.

For example watches are powered using both the kinetic energy of a moving arm and the heat flow away from the surface of the skin. The gradient between the hand and the ambient temperature has been shown to provide a thermoelectric gradient with maximum output current of 18mA and output voltage between 150mV and 250mV.

3.1.3.2 Existing Trends

Passive RFID tags are powered from a high-frequency electromagnetic field generated by the RFID reader. The RFID tag is charging a capacitor that provides enough energy to send back the data stored on the tag. Current research at Intel extends a RFID tag with sensing technology: the sensor data is read out using the normal RFID reader.

Existing solutions for home automation manage to harvest enough energy from the act of pressing a light switch to send on/off commands to the light. Since the light switch has no function during the remaining time, it does not need a battery to power a radio in order to receive messages.

Current research in the field of energy harvesting tries to combine existing techniques to create more efficient power generators, although there is definitely the need to improve the energy generation capabilities of individual techniques. New materials, such as electroactive polymers, are being examined since they promise a higher energy conversion coefficient.

3.1.4 New Sensors and Low-cost Devices

3.1.4.1 Description and Relevance

Real world applications of Cooperating Objects may consist of hundreds to thousands of sensor nodes; therefore unit cost is clearly a market driver. However, for Cooperating Objects to become a commercial success the absolute sensor node recurring cost has to be balanced against the capability it can deliver and hence the through life cost benefit derived. The operational through life costs can include training, downtime/non-availability, maintenance and repair and also the savings such as reduced user interaction, reduced installation costs and benefits derived from continuous monitoring.

The development of Micro-Electro-Mechanical Sensors (MEMS) has led to the production of low-cost sensors. MEMSs provide not only low-cost but also low-powered, low-size sensor nodes. Also production in large volume significantly reduces the cost. As Cooperating Object application areas spread, the need for low-cost sensors increases. Some applications such as environmental monitoring, surveillance, and disaster relief may need to use vast numbers of sensor devices. New efficient and low energy imaging sensors with significant processing capabilities are needed in these applications. Also some sensors in the network may fail due to the environmental conditions and energy constraint that requires the deployment of redundant number of nodes. In such cases, low-cost sensor is a necessity. Otherwise, the implementation of such applications would be hard or even impossible.

3.1.4.2 Existing Trends

There are quite a few sensor device manufacturers in the world that try to reduce the price of sensors in order to make them viable for large-scale deployments. However, the design of new devices is not a cheap undertaking and the lack of standard interfaces makes it even harder for newcomers to enter the market.

Nowadays, the typical sensor node price lies between \$50 and \$200. On the other hand, applications requiring more than 100 sensor nodes dramatically increase investment costs. Silicon-based tilt sensors offer cost effective solutions over older fluid-vial sensors. The ultimate target is to produce sensor nodes with a price of under \$1.

3.1.5 Miniaturization

3.1.5.1 Description and Relevance

Research in the field of Cooperating Objects started with the idea of providing enough computing resources to our everyday lives and to make them more comfortable by the use of computing technologies. These systems are expected to be used in the most varied environments: from inaccessible areas to our own offices or even our own bodies.

Therefore, the sheer numbers of devices that we will have to share our lives with imply that size, biodegradability, etc. play a very important role. Most embedded devices and, of course, sensors too, are designed to be near the source of information they are supposed to monitor. The need for miniaturization arises then by the need to pollute the space as little as possible and to interfere minimally with the observed phenomenon. Additionally, the smaller the devices, the easier it is to carry them around with us or to embed them into the environment seamlessly.

3.1.5.2 Existing Trends

Wireless sensor network devices available today on the market can be considered very small-sized computers. An average sized node in research is around 13 cm^3 (Mica2) or slightly bigger, a small sized mote is around 2.9 cm^3 (Mica2Dot). Despite their small size when compared to traditional systems, they are still too big to be embedded in small objects of daily life, e.g. in smart home scenarios. System on Chip solutions with smaller dimensions exist, but are usually tailored to specific scenarios. Moreover, the key problem is often the battery and packaging and not the sensor node itself.

The vision of Smart Dust is to be able to implement computing devices the size of a grain of sand that will contain sensors, computational power and be able to communicate wirelessly with other devices. Although we are still very far from the realization of this vision, there are already devices available that measure less than 1 cm^3. But there is still the need for small hardware, at an acceptable cost, with enough computational power to implement the vision of smart dust.

3.1.6 Radio Resource Management

3.1.6.1 Introduction

Radio resource management in general deals with the assignment of radio resources to communicating nodes, subject to certain quality or optimization criteria. Some of the most important types of radio resources are:

Spectrum : the amount and center frequency of the spectrum allocated to a communications in general determines either the available data rate or the available redundancy and therefore the reliability. Spectrum is a scarce resource. Exclusive allocation of (portions of) the spectrum to communication partners is beneficial since it eliminates co-channel interference, but reduces the spectrum available for other communicating nodes.

Power : The transmit power is a very important resource, since it influences the achievable signal-to-noise ratio, the transmitter power consumption and the interference created for other stations. It can be varied subject to legal bounds (e.g. the maximum transmit power in the 2.4 GHz band in Europe is 100 mW) and furthermore it depends on the actual type of transmitter whether the transmit power can be varied in a continuous or discrete fashion.

Coding and modulation : by varying coding and modulation (e.g. [53], [218], [71]) one changes the transmission data rate. Typically the trade-off involved here is that higher data rates (achieved with less coding or faster modulations) come at the price of higher residual error rates, which can lead to a higher number of retransmissions when an additional ARQ protocol is used.

Protocols and protocol parameters : by varying certain protocol parameters, another class of trade-offs becomes available. One important example is the variation of the frame length depending on the actual channel (see e.g. [313], [170], [266]), another one the adaptation of the current ARQ scheme to the channel (e.g. in [507] the transmitter uses Goback-N in a good channel state and repetition coding in the bad channel state).

Any adaptive scheme is inevitably tied to channel prediction and the achievable prediction quality (see [104], [519]). The requirements and the effort for channel prediction depend on the involved time scales. For example, when the aim is to closely track the dynamics of fading channels, measurements need to be made with a high frequency, which leads to a prohibitive energy and bandwidth consumption.

In the realm of sensor networks, the variation of coding and modulation is not considered very often (some references are [86], [411]). This can be explained by the fact that the wireless transceivers tend to be very simple in order to save energy, therefore not offering the required capabilities. This is for example true for transceivers compliant to the IEEE

802.15.4 standard, in which the physical layer does not use any modulation or coding adaptation [206]. Furthermore, in many WSN and BSN applications the requested data rates are not very high anyway,

In the remainder of this section we consider mostly the adaptation of power and frequency.

3.1.6.2 Transmit Power Control

Power control is used for many different purposes. If one restricts attention to a single pair of nodes, power control can be used to adjust the received signal power so that a certain target bit error rate is achieved (e.g. [54]). Increasing the power level improves reliability, but excessive transmit power requires too much energy.

In the network case, the situation is more complex, as increasing the transmit power of one station raises the SNR for its intended receiver, but at the same time creates additional interference to other nodes in the network. Without appropriate power control, the other nodes in turn raise their transmit power to improve their SNR and so forth [256].

The power control techniques that are applied in this situation can be categorized as centralized or distributed [168, Chap.3]. In the case of mobile BSNs centralized techniques are applicable within one BSN, but hardly among different BSNs. In the decentralized case it needs to be analyzed, under which conditions a distributed algorithm converges to a network-wide feasible solution. A popular tool for analysis is game theory, see e.g. [438].

Power is often adapted together with other physical-layer control knobs. For example, in [325] power control is combined with coding. Further references are [84], [112], [190]. Another recent trend is to consider power control and power allocation in the context of *cooperative communications* or *cooperative diversity* schemes [129], [93], [187], [338], [408]. In cooperative communications, single-antenna nodes cooperate to provide each other with additional antennas and therefore leverage spatial diversity. Typical approaches in cooperative communications are relaying or cooperative MIMO schemes. In both cases significant work has been spent on the relationship between power allocation and achievable diversity and multiplexing gains, see e.g. [283] [404].

3.1.6.3 Frequency Adaptation

Many BSNs use on the physical layer radios that operate in the ISM bands. As an example, many transceivers for the IEEE 802.15.4 standard operate in the 2.4 GHz ISM band. In these bands, several other technologies are active as well, e.g. WLANs based on IEEE 802.15.4 or Bluetooth. Consequently, the co-existence among these systems is a widely studied subject (see for example [13, 73, 195, 409]). A limitation of most of these studies is that they only consider the case of static networks.

To deal with interfering networks, two different approaches can be used. The first one, followed by Bluetooth and the first version of the IEEE 802.11 standard, is to perform continuous frequency hopping. If the hopping sequence is chosen properly, a sub-band distorted by interference is left quickly towards a better sub-band. Originally, in these networks there was no adaptation, with the newer versions of Bluetooth, however, adaptive frequency hopping was added, a facility for blacklisting consistently bad channels. In contrast, technologies like IEEE 802.11 and IEEE 802.15.4 are designed as frequency-static networks (although for the upcoming version of ZigBee, which operates on top of IEEE 802.15.4, it is expected that frequency-hopping will be included). To adapt the center frequency, the channel quality on the current channel is monitored (e.g. by observing the packet loss rate) and if quality degrades, the system tries to identify a better channel out of the set of channels allowed by the standards. Key questions concern the identification of proper indicators of channel quality, identification of timescales on which they change, and subsequently the design of proper adaptation strategies. In [177] an IEEE 802.15.4-based WSN was used to perform packet loss rate measurements on all 16 channels in the 2.4 GHz range while walking through an urban shopping street. The results indicate that the interference situation changes on timescales in the order of tens of seconds to minutes. This in turn means that to properly perform adaptation, channel state measurements must be available at a time resolution of a few seconds to a few tens of seconds.

3.1.6.4 Interference Mitigation Using Cognitive Radio Techniques

External interferences are a significant source of channel distortions. They are created by other transmitters operating in the same or in neighbored frequency bands. Other transmitters can be stations of the same or different wireless technologies working in the same band, or in industrial environments it could be machinery like arc welders, motors or power electronics.

This problem is especially pronounced in the license-free ISM (industrial, scientific and medical) bands. A range of different technologies share the 2.4 GHz ISM band: IEEE 802.11 WLANs, Bluetooth/IEEE 802.15.1 WPANs, or IEEE 802.15.4 WPANs, and each person is free to buy wireless equipment working in this band and to transmit data (only subject to limitations in transmit power or power spectral density masks). In this situation it is very hard to maintain a given level of QoS over time when the 2.4 GHz ISM band is used – the interference level can for example change due to mobility and the resulting variation in the radio scenery. The co-existence of different technologies on the same frequency band and the resulting performance impacts have been investigated extensively, see for example [13, 73, 195]. It is, however, more attractive to circumvent the interference problem than to live with it.

One possible solution, the allocation of an exclusive frequency band involves interactions with regulatory bodies, like for example the FCC (Federal Communications Commission), and is in general a lengthy and costly process. Another, more immediate solution would be to use other unlicensed frequency bands, like the 5 GHz ISM bands, which are currently less crowded. However, in the higher bands the frequency regulation is not fully harmonized in all countries, and furthermore this only postpones the problem for a while. The concept of *cognitive radio* [16, 97, 130, 178, 311, 523] provides an approach to circumvent this problem.

The concept of cognitive radios is in turn built on the more fundamental concept of *software-defined radios* (SDR) [310], which, roughly speaking, follows the idea to perform (almost) all physical layer signal processing functions in software instead of hardware, for example on a digital signal processor. This gives substantial flexibility, in that it is much easier for software-implemented signal processing algorithms to expose control knobs to higher layers, or to be quickly replaced by other signal processing algorithms with a simple software update. This can for example concern modulation schemes, coding schemes, or center frequencies.

With *cognitive radios*, the idea is to exploit the flexibility of SDR by reconfiguring the radio according to the current state of the wireless terminal and the perceived state of its external environment. In full generality, the environment state can be anything for which the node possesses sensors, more specifically the radio could provide sensing mechanisms to check for the presence of signals in certain frequency bands.

In this context, one of the main uses of cognitive radio is *opportunistic spectrum access*. The motivation behind this idea comes from two observations: (i) electromagnetic spectrum is a scarce resource and license-free spectrum is crowded; and (ii) if a spectrum analyzer is placed at a certain location, one will notice that many exclusively allocated bands are used only intermittently – there are *spectrum holes* which position depends on time and location [178]. A cognitive radio node can exploit these holes: if at his current operating frequency band and position the interference situation degrades, it can sense other frequency bands and seek for new spectrum holes. There is, however, an important constraint: the activity of the cognitive radio nodes must not create any interference to the primary/licensed user when it comes back. To quickly detect the return of the primary user, all nodes in a cognitive radio network must continuously sense the currently used frequency band for the spectral signatures of the primary user and quickly agree on another common center frequency to be used in the future. To minimize disturbance to a primary user, all this should happen within very short time. Instead of specifically seeking for the primary users spectral characteristics, a cognitive radio node could also try to detect the presence of *any* other system in the same band, whether primary or secondary user.

The concept of opportunistic spectrum access can bring significant benefits, since it provides a very promising way to deal with external interferences. Much of the research in opportunistic spectrum access deals with physical layer issues like the development of quick and reliable methods for sensing large portions of the spectrum, methods for detecting the return of primary users etc. With respect to protocols there is a need for signaling protocols within a cognitive radio network to signal presence of interferers to all nodes and to agree on another frequency band. It is conceivable that different networks (e.g. different BSNs) *cooperate* with each other by exchanging measurements or negotiating spectrum usage. Besides the protocols, also suitable *policies* for spectrum sensing (which piece to scan next, which network member performs the scan), for choosing the next frequency and for the cooperation among different networks are worthy research topics.

The IEEE 802.22 working group on wireless regional area networks is currently working on a standard for a wireless medium access control and physical layer that allows opportunistic spectrum access in spectrum that is exclusively allocated to the TV Broadcast Service.[1]

3.2 Algorithms

In this section we present the most representative work on functional properties for Cooperating Objects. By functional properties we refer to the minimum set of algorithms required to run a basic network of Cooperating Objects. Among the areas to be covered in this section are: MAC, Querying, Localization and Data Storage.

3.2.1 Localization

The location of sensor nodes in a deployment area is a very important piece of information required by many applications. The most fundamental requirement is to associate sensor readings with the area where these readings were recorded. Additionally, many types of algorithms rely on such location information, for example, geographic routing algorithms and locality-ware node clustering approaches. As sensor nodes are usually not equipped with GPS or similar localization hardware, node localization has become an important and very active research area for wireless sensor networks.

Recent researches on the integration of Wireless Sensor Networks into higher-level application networks, such as networked robots, raised new challenges for Wireless Sensor Network localization. Thus, when the number of sensors is large, the manual deployment and position recording is error-prone [430] and, in many applications, hand-placing the sensor is not an option. For example, if the sensors are scattered from an airplane, a different localization method should be employed. This is particularly true for networks deployed in emergency response scenarios without preexisting infrastructure, as considered in the AWARE project [19] devoted to the development of a platform for autonomous self-deploying and operation of wireless sensor-actuator networks cooperating with aerial vehicles.

Considering that several Cooperative Objects applications will be resource-constraint, researchers have provided several interesting solutions for different types of localization problems. Next, we present a description of some the most cited work in the area.

[1] http://www.ieee802.org/22/

3.2.1.1 Range-Free

In [48], anchors beacon their position to surrounding neighbors. Using this proximity information, a simple centroid is calculated to estimate location of the non-anchor nodes.

APIT [181] requires a small percentage of nodes (anchors) equipped with high-powered transmitters and location information. Using beacons from these anchors, APIT allows nodes to estimate if they reside inside or outside the triangular region determined by three anchors. By utilizing combinations of anchor positions, the diameter of the estimated area in which a node resides is reduced. The location of the node is estimated by the center of gravity of the reduced area.

Langendoen and Reijers [262] present a comparison of three distributed localization algorithms that share a common structure (Ad-hoc positioning [335], Robust positioning [403], and N-hop multilateration [406]). The evaluation was done on a single simulation platform and the main conclusion is that no single algorithm performs best; their specific performance depends on conditions such as range errors, connectivity and anchor fraction.

3.2.1.2 Range-Based

Range-based technologies for localization have been known for several years. A popular technique (e.g., [489]) infers the distance of a message sender by analyzing the received signal strength indication (RSSI) of received messages. Other important mechanism are: Time of Arrival (ToA) or Time of Flight(ToF), i.e. the transmission time required for radio messages, (e.g., [221]) and Time Difference of Arrival (TDoA) (e.g., [146, 407]) and more recently Angle of Arrival (AoA) (e.g., [326, 335]).

The problem of above distance measurement techniques is their limited precision and their sensitivity to interferences – particularly in scenarios with many obstacles like indoor scenarios. In addition, some of them require of special electronic devices for accurate measurement. Newer approaches aim to overcome these limitations, for example, by measuring distances based on radio interferometry [295] or by using global external events detectable by the sensor chips on the nodes as a source of distance information [257].

Besides classical node localization algorithms based on geographic distances, several projects have developed solutions that rely on external support for assigning coordinates to sensor nodes. The Spotlight localization system uses sensor events for the localization of nodes [445] using a helicopter flying over the deployment area that generates light events. An aerial vehicle is also used together with the RSSI in [124] to tailor a probabilistic framework where the nodes can be localized with errors in the order of one meter. StarDust [446] localizes nodes by recording and analyzing images of the deployment area with light reflected by the individual nodes. The Lighthouse location system [384] determines the distance to the source of a rotating light beam by measuring how long the light sensor of the node is illuminated by the light beam.

As an alternative to geographic coordinates, symbolic coordinates can be used in wireless sensor networks that allow to identify the area nodes are located in. Gauger et al.

[140] discuss different ways of assigning such symbolic coordinates to sensor nodes in indoor scenarios. One idea is to send out coordinate information by broadcast and confirm this triggering a sensor event in the current room by turning on the room light.

3.2.1.3 Target Tracking

Sometimes it is also interesting to localize and/or track objects in the scenario using a Wireless Sensor Network. In [427] the authors present a binary approach to track a single target. The authors propose a filter mechanism to cope with noisy measurements and show through analysis, simulations and empirical evaluation the performance of their approach. In [431], the authors extend this approach to track multiple targets. Simulations and results on a 1D test-bed (chain topology) validates the probabilistic methods presented by the authors.

Some works use Wireless Sensor Networks with nodes equipped with smart cameras to locate or track an object. In [393] it is proposed a real-time cooperative localization and tracking method with a camera-based Wireless Sensor Network that implements an Extended Kalman Filter (EKF). It considers CMUCam2 micro-cameras as sensors integrated in Mica2 nodes that provide measures of the location of the mobile object at a certain time instant. One node collects the readings from all the cameras and implements a burden-optimized EKF to cooperatively locate and track the object.

The variety of applications and user requirements in Cooperating Objects applications has provided a fertile ground to explore new algorithms, and several elegant and functional solutions have been proposed. While the area has been significantly studied, new applications may lead to new research problems in the area.

3.2.2 MAC

The literature on MAC protocols is vast with different goals such a fairness, low power ”consumption” or high throughput. Here we will only review some selected Wireless MAC protocols, designed specifically for WSN, relevant for the context of supporting messages with deadline requirements in wireless ad-hoc networks. Notably, the most significant MAC protocols that achieve this in the context of WSN are designed around some form of TDMA. Undoubtedly, this is not unrelated to the fact that by inherently being collision-free and having the possibility of scheduling transmit/receive times, TDMA-based schemes can be very power efficient.

Common to all TDMA-based protocols is the requirement that nodes have the same time reference. This has been solved in a number of ways. The simplest approach is to use the Global Positioning System (GPS) as the source of a universal clock. GPS can provide extremely accurate timing, but requires special (typically power hungry) receivers and require a clear sky view. Nevertheless, GPS may become standard in designs of sensor network platforms in the near future. Most protocols solve the synchronization problem by transmitting in-band synchronization information. Typically, these involve creating some form of hierarchical organization and use it to distribute timing information. There are several in-band time synchronization schemes in the research literature, where some of the most salient of these, providing good accuracy, are RBS [116], TPSN [136] or FTSP [294]. Notably, the work in [487] is the only practical synchronization strategy that does not require nodes to construct a hierarchical organization, but it can take an unbounded number of broadcasts to achieve synchronization. While researchers have tried to mitigate some shortcomings, often TDMA-based approaches organize nodes in clusters or cells and have a master node proving central coordination, and thus are inflexible to changes in the network topology and the number of participant nodes. Furthermore they have the drawback of requiring that sporadic message streams are dealt with using polling, which is inefficient, specially when the deadline is short, compared to the minimum inter-arrival time of the messages.

TRAMA. The traffic-adaptive medium access protocol (TRAMA) [362] is a TDMA-based MAC protocol that constructs schedules in a distributed manner and on an on-demand basis. It supports both scheduled slots and CSMA-based contention slots for node admission and network management and avoids the assignment of time slots to nodes with no traffic to send. It also allows nodes to determine when they can become idle and not listen to the channel using traffic information. Unfortunately, TRAMA can consume significant computation and memory resources, since it needs to maintain and perform computations upon the two-hop neighborhood list of a node, and this can be very large in dense WSN.

RT-Link. In [389], was developed a hardware platform to support a TDMA protocol that can use an out-of-band synchronization mechanism, avoiding in-band solutions that reduce network performance. In [291], the authors have explored the maximization of parallel transmissions over a TDMA network using RT-Link. This provides optimal end-to-end throughput by identifying the maximal set of concurrent transmitters across the network, while maintaining a bounded delay. However, this result is achieved by assuming that nodes are deployed in a regular structure, something often not applicable in practice.

Implicit EDF. Another approach, Implicit EDF [52], is based on the assumption that all nodes know the traffic on the other nodes that compete for the medium and all these nodes execute the EDF scheduling algorithm. If the message selected by the EDF scheduling algorithm is in the node's queue of outgoing messages then the node transmits this message otherwise it does not transmit. Unfortunately, this algorithm is based on the assumption that a node knows the arrival time of messages on other nodes, thus nodes must to be accordingly placed in static cells, and channel assignment needs to be carefully handled to avoid interference between neighboring cells. This imposes a significant limitation in the real-world applicability of this protocol, and also implies that polling must be used to deal with sporadic message streams.

IEEE 802.15.4. The IEEE 802.15.4 [206] standard covers the physical and MAC layers of a Low-Rate Wireless Personal Area Network (LR-WPAN). It is important to distinguish the IEEE 802.15.4 standard from ZigBee [531]. ZigBee is an industry consortium with the goal of ensuring interoperability between devices. It uses the services provided by IEEE 802.15.4. and defines the higher networks layers and application interfaces to do so. IEEE 802.15.4. was designed for deployment of low-cost, low power wireless networks able to run for years at very low duty cycles. The MAC layer in IEEE 802.15.4 has several operating modes. For the purpose of this section (supporting messages with deadline requirements in wireless ad-hoc networks) the most interesting mode is the beacon-enabled mode, where nodes organize themselves in a Personal Area Network (PAN), and a coordinator (called the PAN coordinator) organizes channel access and data transmissions in a structure called the superframe.

The PAN coordinator is in charged of periodically transmitting a beacon frame announcing the start of the superframe. The superframe is divided in to two main periods: the active period and the inactive period. During the inactive period, nodes in the PAN can turn off their radios, to save energy. The active period is subdivided into 16 time slots, where the first time slot (slot 0) is reserved for the beacon frame. The remaining slots (1 to 15) are used for the Contention Access Period (CAP) and for a maximum of seven Guaranteed Time Slots (GTS). During the CAP, nodes access the medium using slotted CSMA/CA, whereas the GTS is used for reservation-based TDMA access. The GTS slots are allocated by the PAN coordinator, and nodes perform reservation requests during the CAP. A thorough review of IEEE 802.15.4 in the context of supporting messages with deadline requirements in WSN can be found in [246]. A performance study of slotted CSMA/CA can be found in [243], and [247] introduces a mechanism for service differentiation in slotted CSMA/CA by simple manipulation of the protocol's parameters according to the priority of messages. The GTS allocation mechanism was also subject of several studies that address the throughput and delay guarantees provided by this mechanism [244], and energy/delay trade-offs [248]. To overcome the maximum limit of seven GTS allowed, in [245] the authors propose i-Game, an implicit GTS allocation mechanism that enables the use of a GTS by several nodes.

3.2.3 Available Bandwidth Estimation

The estimation of the available bandwidth is an essential task for any resource aware routing protocol. The available bandwidth calculated at the network layer is generally referred to as the effective bandwidth after having considered the channel contention overhead [70]. Available bandwidth estimation and monitoring is one of the essential tasks to accomplish for the development of an efficient methodology for bandwidth management [323]. There have been several proposals in the research literature for the estimation of the available bandwidth, where the wireless channel is generally described as a shared-access communication medium. The available bandwidth varies with the number of nodes contending for the channel and competition for bandwidth is not only end-to-end but also at every link [418].

Three methods are developed in [506] to predict the achievable bandwidth. According to the first method, each node broadcasts its own load information periodically to its one-hop neighbors, in addition to the load information of its two-hop neighbors. This way, each node gathers information on its three-hop neighborhood and uses it for an approximation of the achievable bandwidth. In the second method, the transmission delay is measured, which is inversely proportional to the service rate of the network, which is defined heuristically as the achievable bandwidth. Finally, the third method suggests that the nodes on a defined route also contend with each other and the achievable bandwidth is the minimum available on this route divided by the number of nodes on the route contending with the bottleneck node providing the minimum bandwidth.

The estimation of available bandwidth is considered the basis for admission control. In [419], an admission control and dynamic bandwidth management scheme is proposed. The bandwidth requirement of an application is converted to a channel time requirement and weighted according to the requirements of other connections. The channel time is then shared between connections. The weights are dynamically adjusted as the available bandwidth changes. A central bandwidth manager obtains the bandwidth requirements from the connections at the beginning. It controls admission at connection establishment and redistributes bandwidth shares at connection tear-down. It rejects the connection if the minimum channel time requirement cannot be supported.

Another computation method is developed by the ad hoc QoS on-demand routing (AQOR) algorithm in order to estimate the available bandwidth and perform accurate admission control [503]. Admission control decisions are made by every node based on the analysis of the traffic in the shared channel access network. To this end, each node sends hello packets to its neighbors, which contain information on self-traffic. The total traffic flow in the neighborhood of a node is given as the sum of self-traffic and the traffic of the neighbors, which is deduced from the hello packets received. The available bandwidth is found by subtracting this value from the maximum transmission bandwidth.

As mentioned earlier, these and other bandwidth estimation techniques can be combined with a variety of network layer protocols. Depending on the network conditions and the support of the lower layers, important issues such as contention and interference need to be taken into account. This way, more sophisticated prediction schemes can be integrated into the network layer in order to make the resource reservation decisions more accurate.

3.2.4 Node Clustering

Due to inherent resource constraints in communication and energy consumption, node clustering techniques have been widely utilized by Wireless Sensor Network applications to achieve energy efficiency [511] and scalability. Clustering provides an efficient and scalable network structure for collaborating sensor nodes by grouping them into a hierarchy. Such hierarchical structures are constructed by various clustering approaches at different network layers such as the MAC layer [95, 173, 509, 510] and the routing layer [118, 184, 233, 292].

Clustering offers many advantages in improving the performance of a Wireless Sensor Network. Clustering keeps network traffic local [473] and thus reduces energy dissipation of long-distance transmissions as well as the amount of routing information stored at each sensor node. Clustering can further conserve energy by employing cluster heads (CHs) to perform local data aggregation and activity scheduling among local members. Inactive members can stay in the sleeping mode or low-power operations. Furthermore, clustering also helps in reducing the cost of topology maintenance as a reaction to dynamic topology changes. To be responsive to dynamic phenomenon changes, a collaborative structure needs to be configurable and adaptable [63, 520] to phenomenon dynamics. With a clustered network, topology reconfiguration is only performed on the cluster head level and does not affect local cluster nodes. Thus, the overhead of dynamic topology adaptation can be greatly minimized.

There have been numerous cluster algorithms proposed in the domain of ad hoc networks [7, 29, 165, 231, 478]. Many of these algorithms put their emphasis on node reachability and route establishment. Such topology control [397] approaches address the problem of cluster head selection by finding a *Minimum Connected Dominating Set* (MCDS) [159] or a *Maximum Independent Set* (MIS) [499] for a given Wireless Sensor Network. Blum et al. [37] categorized various MCDS algorithms based on how cluster heads are selected. They also provide a detailed description of these algorithm and an analysis of their performance in terms of number of cluster heads and complexities.

In addition to reachability and route establishment, other critical design goals of WSNs include network coverage and system longevity. The LEACH protocol [183] was among the first cluster-based communication protocols proposed for WSNs. In LEACH, clusters are formed using a distributed algorithm where nodes make local decisions to become cluster heads. LEACH further uses randomization to rotate the cluster heads to balance energy dissipation. LLC [233] introduced a dynamic localized clustering scheme to reduce energy dissipation of cluster heads by adjusting cluster ranges, while the entire WSN is still covered. Instead of adjustable ranges, clusters formed by FLOC [94] are approximately equal-sized and overlapping cluster ranges are minimized. FLOC uses a *solid-disc* clustering property that requires that every sensor node has a unit distance to its cluster head. Ameer et al. [1] survey various cluster algorithms in WSNs and provide a taxonomy and classification of these approaches based on metrics such as convergence rate, cluster overlapping, and on features of clusters such as cluster properties, cluster head capabilities, and cluster process.

To maintain the robustness of Wireless Sensor Networks under stringent resource constraints and high network dynamics, one of the key issues is the development of flexible and adaptive resource management mechanisms that can provide the applications with an abstraction from the physical infrastructure. A cluster-based middleware is presented [512] to achieve this semantic transparency by using a two-layer architecture. According to this, the cluster layer is responsible for forming a cluster from a pool of sensor nodes.

Application information embedded in the application specification is passed down to the cluster layer after being interpreted by the resource management layer. In addition, the cluster layer distributes the commands issued from the cluster head for resource management and cluster control purposes. The resource management layer controls the allocation and adaptation of resources. The resource allocation module within this layer generates an initial solution when the cluster is formed, while the resource adaptation module controls the runtime behavior of the cluster. Using the information by resource management, the cluster head is then responsible for taking adaptation actions. While the cluster forming and control protocol is distributed among all sensor nodes, it is assumed that the code for resource management layer resides at the cluster head. An illustrative technique for energy-efficient resource allocation is also given in [512].

3.2.5 Querying

Querying is perhaps the area that has concentrated most of the interest on Wireless Sensor Network research, and as a result, a number of papers have been published on this topic. In this section, we describe the most representative work for unstructured Wireless Sensor Networks, that is, Wireless Sensor Networks that do not have any knowledge about the network beyond the existence of their immediate neighbors.

Flooding. When there is no *a-priori* knowledge about the event, flooding is the safest alternative, but it leads to serious MAC collisions and redundancy. A seminal work by Ni et al. [332], analyzed the problem and proposed various schemes to alleviate it. These schemes limit the number of nodes broadcasting the query while maintaining a high coverage. Flooding is generally used as the last option for resource discovery, and it has been studied extensively.

Controlled Flooding. When the event is replicated and the probability distribution of the event location is known, controlled flooding (expanding ring searches) can reduce the cost associated with querying [68, 254].

In [199], the authors propose a magnetic diffusion where sinks, acting like magnets, initiate controlled floods with the aim of setting pseudo-magnetic gradients. Then, data is forward greedily through these gradients. The authors report the improvement of Magnetic Diffusion over Flooding-Based querying.

Theoretically, controlled flooding has been studied in different domains, however there have not been empirical tests to accurately assess the performance of this technique

Random Walks. Due to their simplicity and potential low-overhead, random walks have attracted a significant attention as querying mechanisms in the Wireless Sensor Network community. Servetto et al. [415] analyzed the performance of random walks on dynamic graphs (graphs where nodes switch between on and off states) and proposed algorithms to compute local parameters for random walks to achieve load balancing. In [392], the authors present ACQUIRE, where random walks are combined with controlled flooding. The authors show that for some type of queries, ACQUIRE performs better than flooding and controlled flooding.

Rumor routing [43] proposes a push-pull mechanisms, where events and sinks issue random-walk agents the rendezvous. Depending on the frequency of occurrence of events and queries, rumor routing provides a more energy efficient alternative to event or query flooding.

In [17], Avin et al. evaluate the properties of simple random walks on partial cover times. They found that random walks could be more energy-efficient that cluster-head techniques and more robust that spanning trees in the presence dynamics. They also present a modified version of a random walk called biased random walk where, based on a parameter, the walk tends to travel to unvisited nodes. On the same line of work, in [18] the authors present random-walks-with-choice whereas instead of selecting just one neighbor at each step, the walk selects more than 1 neighbor at each step and moves to the node that have been visited the least. The authors show that simple mechanism reduces significantly the cover time and improves load balancing.

While the previous works provide a better understanding on the performance of random walks on Wireless Sensor Networks, most of them are based on ideal communication models. Identifying a random-walk based querying mechanism that exploits that particular characteristics of Wireless Sensor Networks communication graphs is still an open area of research.

3.2.6 Data Processing

Recall from the introduction of this roadmap that Cooperating Objects systems consists of individual entities or objects that jointly strive to reach a common goal. It is expected that Cooperating Objects systems will be of very large scale in the future. The largest system so far is ExScal [12] , a research prototype system comprising more than 1000 nodes. But we can expect even larger systems in the future. It is expected that systems with 10000 nodes will be built in the near future [12]. Also considering the fact that a Cooperating Object system may be comprised of several networks that are owned by different organization, it follows that the number of nodes may be even larger.

Such large networks provide an enormous amount of sensor readings but applications are typically not interested in gathering all sensor readings unless this is strictly necessary or required by law. Applications are mostly interested in obtaining answers to specific queries, most of which involve some form of aggregation, projection or selection of data. Because of this large scale, it is necessary that the time-complexity of performing such queries is small.

3.2.6.1 Data Processing within a Sensor Network

Data processing within a sensor network is typically performed by letting nodes form a hierarchical structure such as (i) a cluster or (ii) a tree. Cluster formation is already

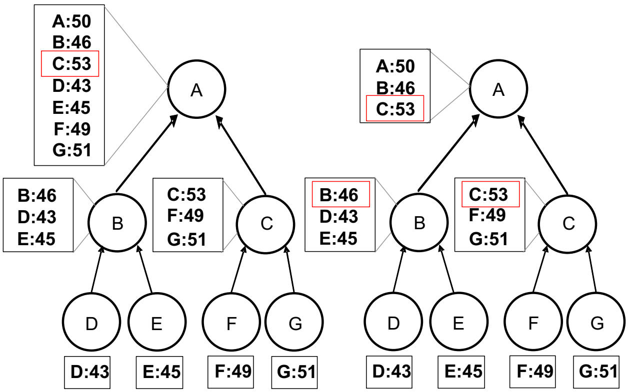

Figure 3.2: *In-Network Aggregation*

discussed in the section 3.2.4. Here we will discuss optimizations that are possible when nodes are organized as a tree.

Distributed query processing [65, 285] is essential for prolonging the lifetime of the network. There are various in-network mechanisms that are widely used in wireless sensor network applications including aggregation, suppression and view management. More recent SNQPs try to achieve query based routing, a technique that incorporates query semantics into query execution. All the aforementioned mechanisms will be thoroughly described in the subsequent text.

In-network aggregation In-network aggregation [10, 284] is the process where results are summarized locally at each sensor when the query executed belongs to the aggregate query class. The sensor device executes local mechanisms that summarize the results according to query semantics and combine the generated results into a single package per query. This process leads to a reduction of packet sizes whilst preserving the quality of data.

To facilitate our description consider the example depicted in Figure 3.2. Figure 3.2 (left) illustrates an example where no in-network aggregation is performed. Sensors record the values generated by their sensing components and forward their results to their parents as soon as the results from all its children are acquired. In our example the query executed is `SELECT MAX(temperature) FROM SENSORS`. The execution starts with sensors D,E,F,G acquire the results 43,45,49,51 respectively and forward them to their parents B and C. B and C record the values 46 and 53 and combine them with the results of their children that is D,E and F,G respectively. This leads to the generation of results $\{B : 46, D : 43, E : 45\}$ at sensor B and $\{C : 53, F : 49, G : 51\}$ at sensor C. Finally, sensors B,C transmit their results to the sink node, A, where the maximum value $\{C : 53\}$ is retrieved.

The aforementioned naive process simply forwards the results of all sensors to the sink node. This leads to the transmission of a large amount of packets which results in additional energy consumption. Figure 3.2 (right) displays the same network configuration but employs in-network aggregation. The execution starts again with sensors D,E,F and G retrieving the values from their sensing components. In the next step they transmit their results to their parent B and C. However, as soon as sensors B,C receive the results, they perform in-network aggregation and produce the values $\{B : 46\}$ and $\{C : 53\}$ which are the maximum values obtained at sensor B and C respectively. As a result they only transmit exactly one result to A. Sensor A, now, has a smaller dataset to process with only three values from itself and B,C. We observe that in-network aggregation leads to a reduction of package size as well as the reduction of processing each sensor locally. This leads to decreased energy consumption of the overall network which is a desired property.

In-network suppression In-network suppression [494] is the process of suppressing (i.e. not transmitting) results when the values of these results lie inside predetermined thresholds defined by the application. This process again leads to decreased communication both with regards to number of transmitted packets and packet size.

To facilitate our description consider the example depicted in Figure 3.3. The sensor network application here continuously records the temperature values of each sensor, to detect temperature anomalies in the topology of the network. The application has set an upper/lower threshold on each sensor so that any detected value falling within the threshold is suppressed and no transmission is necessary. The figure illustrates three subsequent epochs where all sensors record their temperature values. We observe that, for example, sensor D only transmits the value 51 which is recorded in the 3^{rd} epoch as both previous values (49,46) lie within the threshold. Note also that sensors B,C,F do not transmit any values because of this suppression mechanism. This process leads to great energy savings as very few results are transmitted to the sink node. In addition, when the process is used in conjunction with location and time awareness mechanisms, it achieves even greater results.

In-network Caching Similar to in-network suppression, query result caching [517] is the process where results are summarized and cached locally at each sensor. The results are not transmitted if they are exactly the same or if they differ by a minor deviation. This process leads to the decrement of message complexity whilst increasing network longevity.

To facilitate our description consider the example depicted in Figure 3.4. Figure 3.4

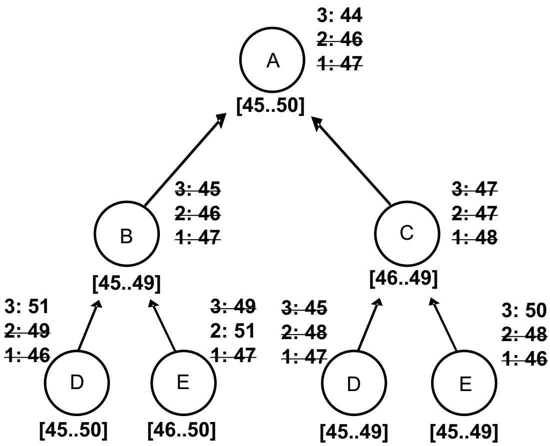

Figure 3.3: *In-Network Suppression*

(left) illustrates the first timestamp of an example where caching is performed. Sensors record the values generated by their sensing components, cache and then forward their results from all its children are acquired. In our example the query executed is `SELECT MAX (temperature) FROM SENSORS`. The execution starts with sensors D, E, F, G acquire the results 43, 45, 49, 51 respectively. They cache them and then forward them to their parents B and C. B and C record the values 46 and 53 and combine them with the results of their children that is D, E and F, G respectively. As soon as they receive the results, they cache them. Then, they produce the values B:46 and C:53 which are the maximum values obtained at sensor B and C respectively. They only transmit exactly one result to A. Sensor A has a smaller dataset to process with only three values from itself and B, C. The values are also cached at sensor A.

Figure 3.4 (right) illustrates the second timestamp of the example. The execution starts with sensors D, E, F, G acquire the results 43, 45, 50, 51 respectively. Because of the fact that the values of the previous timestamp are cached at each sensor, only the values that have been changed need to be transmitted. Thus, sensor F forward its value to its parent C. As soon as sensor C receives the value from its child, produces the value 52. This value needs to be transmitted to sensor A. We observe that if the results are exactly the same then they are not transmitted. This leads to reduced message complexity and increased network longevity.

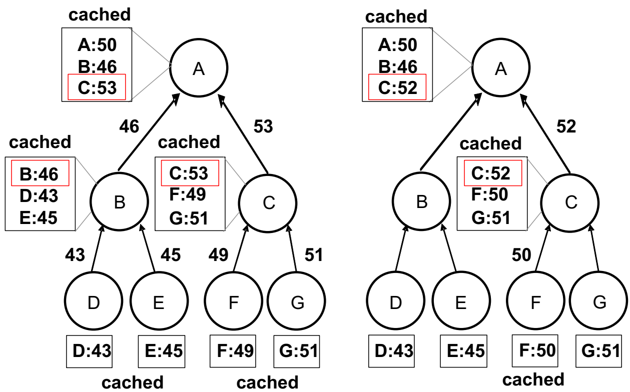

Figure 3.4: *In-Network Caching*

Query-based Routing Query based routing [420, 421] is the process where the network is configured based on query semantics and sensor properties. The process incorporate query semantics into the routing tree construction. This process leads to increased network lifetime, increased network coverage and increased survivability of critical nodes.

To better illustrate the query based routing we use the simple example shown in Figure 3.5. In this figure, nodes 2, 4, and 6 (the shaded ones) belong to one group, whereas nodes 1, 3, 5, and 7 belong to a different group. Let us assume that under the standard First Heard From (FHF) [420] network configuration (Figure 3.5 (left)), nodes 4 and 5 pick 2 as their parent, whereas nodes 6 and 7 pick 3 as their parent. Using in-network aggregation, the message sizes from nodes 2 and 3 to the root of the network will both be 2. On the other hand, if we cluster along the same path nodes that belong to the same group (Figure 3.5 (right)) we reduce the size of messages from nodes 2 and 3 in half: each message will only contain the partial aggregate from a single group.

Resource-Awareness Framework In the attempt to increase energy and communication efficiency, a resource awareness framework for Wireless Sensor Networks is presented [379] that utilizes in-network data processing to adapt to changing resource levels such as battery power, available memory, and computational processing capacity. It is implemented as part of a query processing system, and applied, as a case study, to the query processor's on-line data clustering algorithm. It aims to minimize the cost of data communication by moving data processing algorithms into the sensor networks, and by decoupling

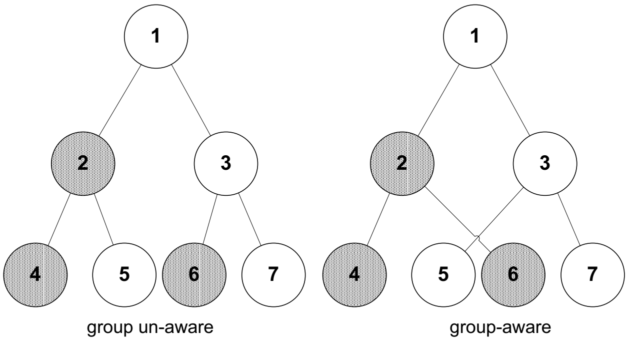

Figure 3.5: *Query-based routing*

data processing and data communication. In the event of high CPU utilization, instead of searching through the entire set of clusters, the algorithm searches only a random subset of clusters, which is a specified proportion of the entire set, to reduce the load on the CPU. In the event of a low free memory level, the algorithm attempts to reduce its memory footprint by reducing the size of the set of clusters. The framework can keep a constant memory footprint for only a marginal acceptable error in result accuracy.

3.2.6.2 Data Processing within a Broadcast Domain

We say that the set of nodes N is in a single broadcast domain if the following holds for every every node $N_i \in N$: if N_i broadcasts a packet or an energy pulse then it holds that every node in N (except possibly N_i) will correctly receive this transmission. The fact that all nodes are in a single broadcast domain can be exploited to achieve scalable data processing when a prioritized MAC protocol is used. We will discuss how to do this.

We will first see how dominance protocols work; this is a certain class of prioritized MAC protocols that offer a very large number of priority levels. After that, we will see how such a protocol can be used to compute minimum of sensor readings. This gives the main idea on why prioritized MAC protocols are useful for scalable data processing. Finally, we will list other computations that can be performed based on the same idea.

Preliminaries and Motivation The basic premise for our discussion is the use of a prioritized MAC protocol. This implies that the MAC protocol assures that out of all nodes contending for the medium at a given moment, the one(s) with the highest priority

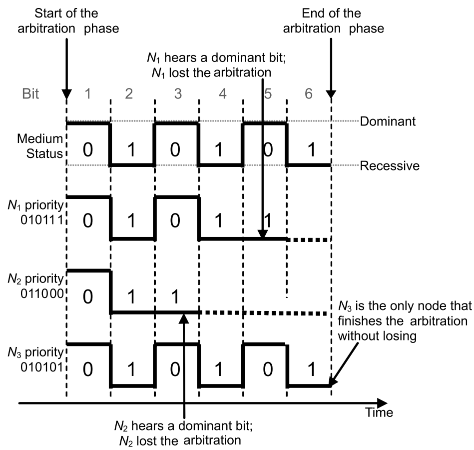

Figure 3.6: *Prioritized MAC protocol; illustration of Dominance/Binary-Countdown Arbitration*

gain access to it. This is inspired by Dominance/Binary-Countdown protocols [315]. In such protocols, messages are assigned unique priorities, and before nodes try to transmit they perform a contention resolution phase named arbitration such that the node trying to transmit the highest-priority message succeeds.

During the arbitration (depicted in Figure 3.6), each node sends the message priority bit-by-bit, starting with the most significant one, while simultaneously monitoring the medium. The medium must be devised in such a way that nodes will only detect a '1' value if no other node is transmitting a '0'. Otherwise, every node detects a '0' value regardless of what the node itself is sending. For this reason, a '0' is said to be a dominant bit, while a '1' is said to be a recessive bit. Therefore, low numbers in the priority field of a message represent high priorities. If a node contends with a recessive bit but hears a dominant bit, then it will refrain from transmitting any further bits, and will proceed only monitoring the medium. Finally, exactly one node reaches the end of the arbitration phase, and this node (the winning node) proceeds with transmitting the data part of the message. As a result of the contention for the medium, all participating nodes will have knowledge of the winner's priority.

The CAN bus [40] is an example of a technology that offers such a MAC behavior. It is used in a wide range of applications, ranging from vehicles to factory automation

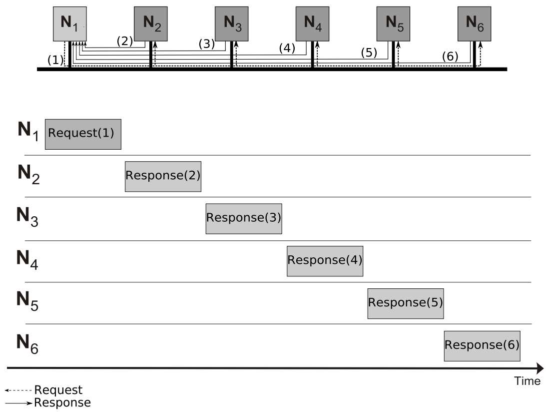

Figure 3.7: *Naïve algorithm (TDMA-like MAC)*

(the reader is referred to [56] for more examples of application fields and figures about the use of CAN technologies). Its wide application fostered the development of robust error detection and fault confinement mechanisms, while at the same time maintaining its cost effectiveness. An interesting feature of CAN is that the maximum length of a bus can be traded-off for lower data rates. It is possible to have a CAN bus with a bit rate of 1Mbit/s for a maximum bus length of 30 meters, or a bus 1000 meters long (with no repeaters) using a bit rate of 50 Kbit/s. While the typical number of nodes in a CAN bus is usually smaller than 100, with careful design (selecting appropriate bus-line cross section, drop line length and quality of couplers, wires and transceivers) of the network it is possible to go well above this value. For example, CAN networks with more than a thousand nodes have been deployed and they operate in a single broadcast domain (such networks have been built; see for example [234]).

WiDom [349] is another example of a technology that offers such a MAC behavior. WiDom is a recently proposed research prototype but unlike CAN, it can operate on wireless channels.

The main idea The problem of obtaining aggregated quantities in a single broadcast domain can be solved with a naïve algorithm: every node broadcasts its sensor reading sequentially. Hence, all nodes know all sensor readings and then they can obtain the aggregated quantity. This has the drawback that in a broadcast domain with m nodes,

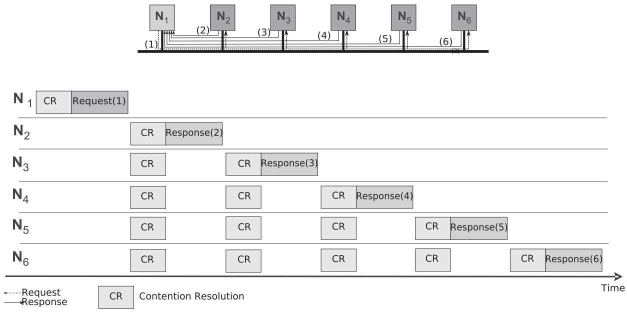

Figure 3.8: *Naïve algorithm (Prioritized MAC)*

at least m broadcasts are required to be performed. Considering a network designed for $m \geq 100$, the naïve approach can be inefficient; it causes a large delay.

Let us consider the simple application scenario as depicted in Figure 3.7, where a node (node N_1) needs to know the minimum (MIN) temperature reading among its neighbors. Let us assume that no other node attempts to access the medium before this node. A naïve approach would imply that N_1 broadcasts a request to all its neighbors and then N_1 would wait for the corresponding replies from all of them. As a simplification, assume that nodes orderly access the medium in a time division multiple access (TDMA) fashion, and that the initiator node knows the number of neighbor nodes. Then, N_1 can derive a waiting timeout for replies based on this knowledge. Clearly, with this approach, the execution time depends on the number of neighbor nodes (m). Figure 3.8 depicts another naïve approach, but using a prioritized MAC protocol.

Assume in that case that the priorities the nodes use to access the medium are ordered according to the nodes' ID, and are statically defined prior to runtime. Note that in order to send a message, nodes have to perform arbitration before accessing the medium. When a node wins it sends its response and stops trying to access the medium. It is clear that using a naïve approach with a prioritized MAC protocol brings no timing advantages as compared to the other naïve solution (Figure 3.7).

Consider now that instead of using their priorities to access the medium, nodes use the value of its sensor reading as priority. Assume that the range of the analog to digital converters (ADC) on the nodes is known, and that the MAC protocol can, at least, represent as many priority levels. This assumption typically holds since ADC tend to have a data width of 8, 10, 12 or 16-bit while the CAN bus offers up to 29 priority bits. (And WiDom can be configured with the same number of priority bits if needed.) This alternative

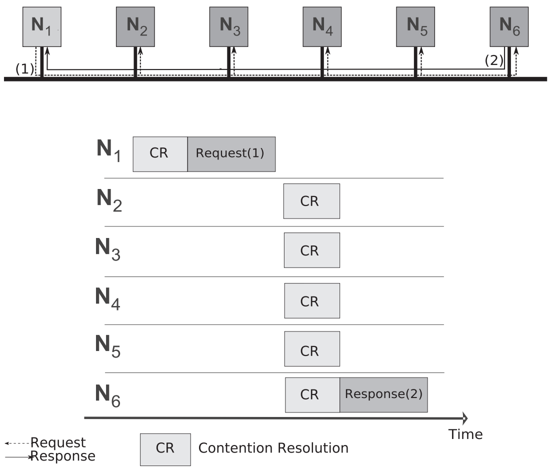

Figure 3.9: *Innovative algorithm (Prioritized MAC)*

would allow an approach as depicted in Figure 3.9. With such an approach, to obtain the minimum temperature among its neighbors, node N_1 needs to perform a broadcast request that will trigger all its neighbors to contend for the medium using the prioritized MAC protocol. If neighbors access the medium using the value of their temperature reading as the priority, the priority winning the contention for the medium will be the minimum temperature reading. With this scheme, more than one node can win the contention for the medium. But, considering that at the end of the arbitration the priority of the winner is known to all nodes, no more information needs to be transmitted by the winning node. In this scenario, the time to obtain the minimum temperature reading only depends on the time to perform the contention for the medium, not on m. If, for example, one wishes that the winning node transmits information (such as its location) in the data packet, then one can code the priority of the nodes by adding a unique number (for example, the node ID) in the least significant bits, such that priorities will be unique. Such use, results in a time complexity of $\log m$.

A similar approach can be used to obtain the maximum (MAX) temperature reading. In that case, instead of directly coding the priority with the temperature reading, nodes will use the bitwise negation of the temperature reading as the priority. Upon completion of the medium access contention, given the winning priority, nodes perform bitwise negation again to know the maximum temperature value.

MIN and MAX are just two simple and pretty much obvious examples of how aggregate quantities can be obtained with a minimum message complexity (and therefore time complexity) if message priorities are dynamically assigned at runtime upon the values of the sensed quantity.

Other computations Logical-OR [376] among boolean values distributed on different nodes can be computed efficiently using MAC layer support. The main idea is that nodes agree on a common time interval and synchronize their clocks. If the boolean value on a node is "true" then the node broadcasts an unmodulated carrier and this node knows that the result of the logical-OR is clearly "true". If the boolean value on a node is "false" then the node performs carrier sensing; if it heard a carrier then it knows that the result of the logical-OR is "true", otherwise it is "false". It can be seen that this computation can be interpreted as a special case of the MAX computation above by setting the number of priority bits equal to one.

Although the technique for evaluating the function logical-OR [376] is very simple, it has great potential because logical-NOR can be computed from logical-OR and any logical function can be computed from logical-NOR.

Recall that in the ideal world, all nodes should have knowledge of sensor readings everywhere but this is expensive in terms of communications. One practical approach is to let nodes obtain an approximate representation of all sensor readings. An interpolation of space coordinates is appropriate for this purpose because sensor readings typically vary "slowly" with changes in the space coordinates. An approach that exploits a prioritized MAC protocol can be used to obtain such an interpolation [8]. The main idea is to start with an initial interpolation (this can be very simple, for example zero everywhere) and then let each node N_i compute its error, that is, the difference between the interpolated value at N_i as compared to the real sensor reading at N_i. Then nodes assign their priorities being equal to this difference and the MAC protocol selects the node with the largest priority (that is largest error). This node broadcasts its sensor reading and its position and then all nodes update the interpolation function to take this new information into account. This procedure is repeated until all nodes have a sufficiently small error.

A prioritized MAC protocol can also be used to estimate the number of nodes. This is useful in its own right in order for nodes to know how many nodes are alive after catastrophe (for example an earthquake). But its main usefulness is as a building block in other algorithms, for example majority voting and hypothesis testing. Its use in hypothesis testing can be seen as follows. Let us consider the case that a sensor network desires to know if a certain hypothesis about the physical environment is true (for example "There was a sniper at position (x,y,z) who shot a bullet at time t.). This can be achieved by letting each node take sensor readings (for example acoustic signals) and based on those, decide (based on some error margins) if the hypothesis is compatible with the sensor readings the node has taken. We can then count the number of nodes for which the hypothesis is compatible with its sensor readings. If this number of nodes is sufficiently high then the hypothesis can be trusted. Note that this approach can also be used to find the hypothesis that is most compatible with sensor readings of all nodes.

3.2.7 Cooperation of Moving Autonomous Vehicles and Robots

3.2.7.1 Description and Relevance: Motivation of the Research

The main motivation is the consideration of many mobile objects (vehicles) interacting and cooperating in the same physical space. This motivation involves several topics related to coordination and cooperation paradigms, reliability and safety. Furthermore, security and safety requirements impose the consideration of faulty or malicious objects (vehicles) in the same space.

In the following we analyze this motivation and the relevance of the different topics involved.

The application of distributed methods for spatial and temporal coordination of mobile objects is relevant to provide reliability and scalability in missions that require mobile objects sharing the same space.

Behavior based intrusion detection is also a relevant topic for safe motion in shared space in the presence of faulty and malicious vehicles (reactive measure).

The application of fully distributed mission planning and task allocation methods is relevant to deal with the spontaneous ad-hoc cooperation in missions with large numbers of entities that have to interact, and also in network centric computing where direct communication with a central station is not possible. It is also relevant if the local dynamics of the situation require timely reaction of the entities involved in it.

Methods to compute the optimal coverage are needed in missions requiring the maximization of the covered area and the absence of internal unsensed regions.

Sensing and perception methods and technologies are also needed for safe motion and navigation in dynamic uncertain environments, detecting static obstacles and mobile objects.

The cooperation of the mobile objects also requires the integration of individual environment perception of the mobile objects to achieve a better environment model. The so-called active perception methods are used to generate motions and other actions that improve the environment perception.

Moreover, control methods and technologies are also required when then cooperation involves the generation of actions and particularly when there are physical interactions with the environment, as for example in the cooperative manipulation of objects in the environment.

Middleware for highly mobile environments is also a need in the cooperation of mobile objects. This middleware should provide suitable communication paradigms and unified distributed data delivery without central control for the considered mobile scenarios.

Finally, the cooperation of mobile objects also requires methods to improve communication by means of distributed decision making that optimize communications in poor/insufficient radio coverage conditions. These distributed decision making methods are also relevant to be able to maximize the network performance, reacting to changes in the information flow with as little global information as possible.

3.2.7.2 State of the art: Basic Concepts Involved

The research on the mobility of Cooperating Objects and particularly the cooperation of multiple mobile objects involves a number of basic concepts and theories that have been developed in different areas.

Thus, several methods developed in mobile robotics such as position estimation under uncertainties, collision avoidance, trajectory optimization and motion planning can be used.

Decision making theories are relevant in mission planning, task allocation and intrusion detection. These include byzantine agents, team utility maximization, distributed negotiation protocols and optimal assignment.

The basic concepts involved in sensing and perception are data fusion, optimal deployment under limited sensing for coverage, co-operative perception, rumor propagation and active perception.

New control approaches of Cooperating Objects interacting with the environment are also related to environment perception and interpretation, and self-monitoring to improve reliability.

The involved concepts and methods in communication are data centric approaches, intelligent routing, distributed negotiation protocols for communication, discovery and collaborative assignation in relay tasks.

Furthermore the integrity, authenticity and confidentiality in mobile object cooperation are based on the message integrity and authentication, and the software stack integrity.

All the above concepts and methods have been used in the mentioned areas, i.e. robotics, control, decision making, and communication. However, their integrated application for the coordination of mobile objects sharing the same physical space in cooperative missions is still in its infancy.

3.2.7.3 Trends: Main Approaches

The following promising approaches have been identified:

- Scalable optimal methods for collision avoidance by using discrete and continuous representations of the space, including velocity and trajectory planning.
- Byzantine agreements for behavior based intrusion detection
- Market based approaches for negotiation in fully distributed mission planning and task allocation.

- Optimal coverage by using distributed policies for sensor deployment and Voronoi tessellation.
- Sensing and environment perception by means of distributed data fusion, sense and avoid, fault detection, and multi-vehicle reliability.
- Perception for cooperation by means of probabilistic data fusion and entropy based control
- Control methods with novel sensor data fusion, new control techniques and environment perception in the control loop,
- Middleware with publish/ subscribe mechanisms for unified data exchange, support for commands to better support actuations, and gossiping based routing approaches
- Communication with QoS estimation and measurement, negotiation, and distributed routing algorithms with special focus on location aware routing techniques.
- Integrity, authenticity and confidentiality by using cryptography with message authentication code and digital signatures, key distribution/revocation and remote attestation.

3.2.7.4 Enabling Technologies

The practical application of the methods and technologies developed in Mobility of Cooperating Objects will require technologies that are not considered within it.

Particularly, new affordable mobile communication technologies improving current range and band are required. This is particularly true when considering reliable communication between fast mobile objects in broad areas.

Another technology demand is new light, affordable and reliable sensors for mobile object detection, including 3D scanning sensors. Furthermore, embedded sensing for mobile object position estimation in GPS denied environments is also a need. Moreover, applications such as environmental monitoring also require new sensors.

Cooperative human-machine approaches for perception and mission execution have been also identified as required technologies.

3.2.7.5 Analysis of Applications: Short Term

Environmental monitoring by means of a fleet of vehicles is a suitable scenario for short term applications. This scenario is particularly interesting when considering broad areas with minimal infrastructure. Aerial vehicles are particularly valuable when considering ground locomotion constraints in most environmental scenarios.

Another possibility is emergency and disaster scenarios, particularly without sensing and communication infrastructure. These scenarios involve not only exploration and detection of events but also tracking of mobile objects.

The application of autonomous vehicles and mobile robots is interesting in the above scenarios and others related to monitoring and filming activities.

3.2.7.6 Analysis of Applications: Long Term

The future air traffic control and management systems require the development of new network based methods for coordination of mobile objects involving many of the above characteristics including scalability, real-time properties, uncertainty management, safety and security.

Another future application area is the multi-UAV coordination, including environment manipulation (e.g. equipment installation) for environmental monitoring, emergency and other scenarios.

The insertion of UAV in non-segregated aerial spaces has been recognized as one of the main issues for the UAV civil applications. Furthermore, new UAV air traffic management systems should consider the integration of UAVs. These applications demand the development of new mobile cooperating methods and technologies.

3.3 Non-functional Properties

3.3.1 General Aspects

Non-functional Properties (NFPs) are defined as the properties of a system that do not affect its functionality, but its quality. We consider NFPs as the Quality-of-Service (QoS) characteristics of a system, where QoS should be interpreted in a holistic way, instantiated in properties such as energy-efficiency /system lifetime, reliability/robustness (processing, communication, radio links, sensors/calibration), timeliness (throughput, delay, traffic differentiation real-time/best-effort), availability, maintainability, safety, security, scalability, mobility, heterogeneity and cost (see Figure 3.10).

According to each application task requirements, which can be rather diverse ([366]), computations and communications must be correct, secure, produced “on time” and with the smallest energy consumption possible. Cooperating Objects systems must also be cost-effective, maintainable and scalable. All this intends to stress the fact that all NFPs are interdependent, in a way that changing one of them may impact other. Just as an example, implementing more reliable Cooperating Object systems may increase cost (structural redundancy, e.g. hardware replicas), reduce energy-efficiency/system-lifetime (e.g. information redundancy, e.g. adding parity or CRC/FCS bits to messages) or reduce throughput (time redundancy, e.g. repeating a computation or resending a message).

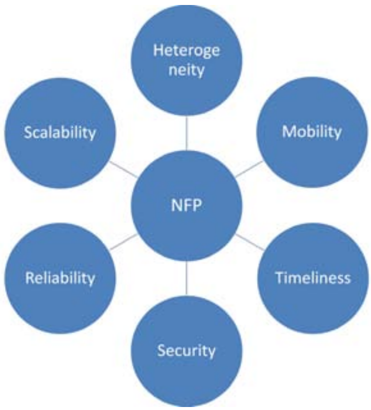

Figure 3.10: *Non-functional properties*

[493] presents an excellent survey on the non-functional properties (i) real-time and (i) reliability of wireless communications. For this first version of the CONET roadmap, we will elaborate on the NFPs that were defined in the Embedded WiSeNts roadmap ([296], Sections 1.3 and 7.1), instantiating in: Scalability, Timeliness, Reliability/Robustness, Mobility, Security, Heterogeneity. The following sub-sections outline the current state of research, practice and technology concerning these NFPs.

3.3.2 Scalability

3.3.2.1 Description and Relevance

A Cooperating Objects system may involve different entities, such as network nodes (for serving as sensors/actuators, routers/ gateways and/or sinks/controllers), machines (e.g. roller belt, mobile robot, fridge, traffic light) or living creatures (plants, animals, humans, bacteria). Depending on characteristics such as the application, the environment or the users, a Cooperating Objects system scale may dynamically change with time. The term "scale" applies to the number (fewer or more nodes in the overall system), spatial density (fewer or more nodes in a restricted region), or geographical region under coverage (smaller or wider, 2D or 3D). The ability of a Cooperating Objects system to easily/transparently adapt itself to these dynamic changes in scale is named "scalability".

Consider an application used for early detection of forest fire which is implemented in a huge forest such as the Amazons. Depending on the sensing information granularity (more sensor density leads to richer information, but also to more information to transmit and process) that is required and to the very limited transmission range of WSN nodes (few meters), the network may scale up to thousands nodes in order to cover the whole area. In such a case, the algorithms running inside the network should scale well in parallel to the increasing number of nodes in a region, still guaranteeing that the application behaves correctly. Additionally, the system should adapt itself to these scale changes in a transparent way, i.e. without requiring user intervention.

Note that while it might be straightforward that scalability is an important issue for "outdoor" applications, "indoor" applications such as factory automation, security and domotics might also impose a high level of scalability to the underlying Cooperating Objects system.

3.3.2.2 State of the Art

To the authors' best knowledge, while some ongoing efforts envisage to effectively build WSNs with hundreds/thousands of sensing nodes (e.g. VigilNet, [180]), the ExScal project (Elements of an Extreme Scale Wireless Sensor Network, [12]) engineered the largest Wireless Sensor Network test-bed so far. A 1000+ node Wireless Sensor Network and a 200+ node peer-to-peer ad hoc network of 802.11 devices were deployed in a 1.3 km by 300 m remote area in Florida (USA), late 2004.

Although a very large number of processors and sensors can operate in parallel and hence the processing and sensing capabilities increase linearly with the number of sensor nodes, the communication capability does unfortunately not increase linearly with the number of sensor nodes. Consider for example 1 million WSN nodes densely deployed in a small area. Two nodes sending simultaneously would cause a collision and hence it is necessary that at most one node sends at a time. With typical WSN nodes today, it takes at least 1 ms to send a message, and hence it takes at least 1000 seconds (approximately 20 minutes) for all nodes to send their data. In dynamic environments subject to rapid changes with time (which is typically the case in Cooperating Objects systems), this might be unacceptable, or at least undesirable. It is also unacceptable from an energy-efficiency perspective because all nodes need to be "awake" for all these 20 minutes just to compute an aggregated quantity (say minimum temperature) from the sensor readings.

Therefore, it is of particular importance that the communication protocol (or protocols) serving as the networking infrastructure for Cooperating Objects systems are designed with scalability in mind. For instance, Medium Access Control (MAC) and routing mechanisms must encompass scalability, otherwise problems such as uncontrolled medium access/routing delays or routing tables' buffer overflows may occur. Scalability must also be taken into consideration for achieving efficient data processing, aggregation, storage and querying in Cooperating Objects systems, especially when large amounts of data are involved. Recent findings on wireless dominance-based MAC protocols (like the one used in the Controller Area Network [151]) provide unprecedented advantages for Wireless Sensor

Networks, namely because aggregate computations can be performed with a complexity that is independent of the number of sensing nodes [8]. Currently the approach is capable of (i) computing the maximum of sensor readings on all sensing nodes, (ii) computing the minimum of sensor readings on all sensor nodes, (iii) obtaining an interpolation as an approximate representation of all sensor readings, (iv) obtaining an estimate of the number of sensing nodes and (v) iteratively search for a hypothesis that is compatible with the sensor readings that the majority of sensor nodes had.

One strategy towards a better support of network scalability relies on the use of hierarchical (or tiered) network architectures. Cluster-based (e.g. [531], [162] or [183]), hexagonal ([359]) or heterog neous-protocols (e.g. [250], [152]) are some examples. In the latter case, the communication architecture is composed of a more powerful (e.g. higher energy capacity, radio coverage and bit rate) network technologies serving as a backbone to less powerful (sub)networks at the sensor/actuator level.

Several research works and commercial products propose hierarchical architectural solutions for Wireless Sensor Networks, namely for enabling Internet to get into the "smart objects" level. The concept of multiple-tiered network architectures has been employed since a long time ago in other networking domains (e.g. Switched Ethernet over field-bus networks in industrial environments or Internet (IP) running over different lower level protocols - ATM over Switched Ethernet).

[241] proposed the use of a two-tiered WSN architecture for structural health monitoring . This is a GSM-like architecture that divides the monitored area into several clusters. Each cluster is managed by a local master that handles the communication using a TDMA-like protocol inside the cluster. This approach lacks scalability inside each cluster due to the TDMA inherent limitations. Also, this architecture is entirely dependent on the presence of a local master to ensure communications, which is not suitable for WSNs. In fact, for a large-scale network, this architecture is unpractical since the number of local master's increases linearly with the number of deployed nodes, resulting in a significant increase of the overall cost.

[152] proposed using a gateway as a portal where every Wireless Sensor Network node is identified by an IP address, allowing direct and individual access. However, there is no mobility support and the handling of very large networks may become a difficult task. [250] and [263] propose a multiple-tiered architecture relying on a IEEE 802.11/WiFi-based backbone and a IEEE 802.15.4/ZigBee-based sensor/actuator network. Though there is a concern on supporting QoS in IEEE 802.15.4/ZigBee-based Wireless Sensor Networks, especially on supporting both best-effort and real-time traffic, there are still lots of open issues to be solved, specially at the backbone network level.

Some commercial solutions rely on IP/Ethernet for their backbone network. These approaches might be cost effective and reliable for small and static networks but the scalability for the higher tier (IP/Ethernet) is limited by the need of a physical Ethernet port for every gateway. Additionally, other QoS features (such as timeliness) are basically neglected.

We have seen that two approaches can be used for achieving scalability (i) computations using a prioritized MAC protocol and (ii) hierarchical structures. These approaches can work in synergy; subdivide the network into clusters and create a hierarchy of clusters and use the prioritized MAC protocol in each cluster [9].

3.3.3 Timeliness (Throughput, Delay and Real-time)

3.3.3.1 Description and Relevance

The ubiquity and pervasiveness of Cooperating Object systems will lead to a very tight integration and interaction between embedded computing devices and the physical environment, via sensing and actuating actions [441]. Such "cyber-physical" systems require a rethinking in the usual computing and networking concepts, and given that the computing entities closely interact with their environment, timeliness is of increasing importance [442]. The "timeliness" NFP concerns the timing behavior of a system, including issues such as network throughput (effective bit rate) and transmission delay (how long does it take for a message to be transmitted from source to destination).

Some Cooperating Object applications, or some specific tasks within an application, might also impose to be finished within a certain time limit (deadline). In this case, we usually refer to these as "real-time" applications/tasks, encompassing the need for real-time computation (requiring real-time operating systems and programming languages) and real-time communications (requiring real-time communication protocols). For instance, in a Cooperating Object system there might be a task that is to process a certain event (e.g. gas leak) in a certain region and transmit that information to a remote sink within 10 seconds (at the latest). Note that the timing behavior or Cooperating Object hardware, such as sensors/actuators, signal conditioning circuits and analogue-to-digital converters, must also be considered due to its impact in monitoring/control loops.

Usually, two classes are distinguished, namely hard real-time applications and soft real-time applications. Hard (or strict) real-time means that missing a deadline leads to a critical or catastrophic failure in the application domain; hence, temporal constraints must be strictly respected to ensure the reliable operation of the application. Examples of hard real-time application are the ABS car breaking system or the control of a manufacturing robot. Soft real-time means that the application can survive or tolerate missing some deadlines, just leading to a "quality degradation"; a typical example would be multimedia streaming over a network. A soft real-time system tries to minimize the deadline miss ratio, or to provide a probabilistic guarantee on the deadline miss ratio.

The general principle of real-time systems design is to ensure temporal predictability of the tasks involved in the application, and in their scheduling. Hard real-time systems require a strict worst-case execution time (WCET) analysis of the tasks (and the related worst-case transmission times for the communication aspect), while soft real-time systems can use statistical analysis based on code profiling, simulation or real experiments.

A fundamental difficulty in designing Cooperating Object systems with real-time requirements results from design principles that are usually antagonist to "traditional" real-time systems. "Traditional" real-time systems require over-allocation of resources (resulting from the inherent pessimism of the analysis, e.g. WCET), usually reducing their adequateness to tackle the dynamic behavior of the physical phenomena. On the other hand, Cooperating Object systems, which rely mostly on unattended resource-constrained WSN nodes, try to optimize resource usage, and also depend heavily (by definition) on the dynamic nature of their environment. An example is tracking the motion and evolution of a fluid (e.g. gas leak) with a Cooperating Object system, where the computational and communication demands change in time and space, according to the propagation of that fluid.

The hidden-node (or hidden-terminal) problem has been shown to be a major source of QoS degradation in WSNs, due to factors such as the limited communication range of sensor nodes, link asymmetry and the characteristics of the physical environment. In wireless contention-based MAC protocols, if two nodes that are not visible to each other transmit to a third node that is visible to the formers, there will be a collision (usually called hidden-node or blind collision). This problem leads to the degradation of three performance metrics: throughput, which denotes the amount of traffic successfully received by a destination node and that decreases due to additional blind collisions; energy-efficiency, that decreases since each collision causes a new retransmission; message transfer delay, which represents the time duration from the generation of a message until its correct reception by the destination node, and that becomes larger due to the multiple retransmissions of a collided message.

3.3.3.2 State of the Art

As already referred in 3.3.1, all NFPs are interdependent. This also applies to timeliness, meaning that, for instance, to increase network throughput we might opt for increasing the "hardware" bit rate or increasing the WSN nodes' duty cycle, which both lead to more energy consumption.

Real-time issues have only recently drawn attention from the Cooperating Objects and Wireless Sensor Network scientific community ([442]). However, the real-time behavior of Cooperating Object systems will be of increasing importance for many applications: real world processes and phenomena often require real-time data acquisition and processing ([441]). Some examples include mission critical applications, such as early warning systems for natural disasters or contamination (forest fires, earthquakes, tsunamis, radiation, etc.) or support for emergency interventions (firemen, etc.). Real-time constraints may be even more stringent in applications such factory automation, health case, ambient assisted living or intelligent transportation systems.

In this context, it is crucial that WSN resources are predicted in advance, to support the prospective applications with a predefined timeliness. Thus, it is of paramount importance to have adequate methodologies to dimension network resources in such a way that the system behaves as expected [442]. However, the provision of timeliness guarantees has always been considered as very challenging due to the usually severe limitations of WSN nodes, such as the ones related to their energy, computational and communication capabilities, in addition to the large-scale nature of WSNs. So, adequate mechanisms must be devised for dimensioning WSN resources so that to guarantee a minimum timeliness performance.

Actually, the evaluation of the performance limits of WSNs is a crucial task, particularly when the network is expected to operate under worst-case conditions [198]. For achieving real-time communications over sensor networks, it is mandatory to rely on deterministic routing and MAC (Medium Access Control) protocols. Usually, these networks use hierarchical network / topological models such as hexagonal, grid or cluster-tree (e.g. [3], [149], [359], [250], [217]). Basically, these network models rely on (1) the use of contention-free MAC protocols (e.g. (i) Time Division Multiple Access (TDMA) or (ii) token passing or (iii) strictly prioritized MAC protocols and unique priorities [350]) to ensure collision-free and predictable access to the medium, and (2) the ability to perform end-to-end resource reservation. These represent important advantages of hierarchical topologies when compared to what can be achieved in flat mesh-like topologies, where contention-based MAC protocols and probabilistic routing protocols are commonly used, preventing them from providing a deterministic performance (timing and buffer).

Concerning the mitigation of the hidden-terminal problem, several mechanisms have been proposed, but mostly addressing "traditional" wireless networks. The can be grouped into the following categories: busy tone mechanism (e.g. [454], [498] and [164]); RTS/CTS mechanism (e.g. [455], [224], [204], [505]; Carrier Sense Tuning (e.g. [96], [501], [518]); Node Grouping [203].

3.3.4 Reliability/Robustness

3.3.4.1 Description and Relevance

In Cooperating Object systems, the operational and environmental conditions may be unfavorable, particularly the ones relying on Wireless Sensor Networks [64, 495, 522, 527]. Generally speaking, Wireless Sensor Network devices such as sensors, actuators, etc., should be resistant to potentially harsh environmental conditions such as vibration/mechanical impacts, high and/or low temperature, water/humidity/moisture, dust or other RF devices or Electromagnetic Interference (EMI) sources. Some Cooperating Object applications, such as environmental monitoring, security and surveillance, and disaster relief, may be deployed in hostile environments and need to have a lifetime of several years. Underground (e.g. mines, metropolitan) and underwater (e.g. tsunami detection or animal monitoring) deployments are also quite challenging in terms of hardware robustness.

Robustness (hardware/software) refers to a component or system that performs well not only under ordinary conditions but also under abnormal conditions that stress its designers' assumptions. Reliability is the ability of a system or component to perform its required functions under stated conditions for a specified period of time. This is especially important since, in many Cooperating Object applications, once the nodes are deployed it may be extremely difficult or even impossible to access them again (e.g. for maintenance or upgrading purposes). In such applications, nodes are expected to live as long as possible. To achieve these high levels of reliability, it is important that Cooperating Object systems can be both robust and support fault-tolerance mechanisms.

Also algorithms used throughout the Cooperating Object system must be resistant and adaptive to sudden and/or long-term changes, that is, apart from the hardware, the software must also be robust. An algorithm/protocol is robust if it continues operating correctly despite abnormalities (e.g. in inputs, calculations). Algorithms used for routing, localization, mobility, etc., should keep working properly even if operational conditions or the structure of the system/network change.

Data delivery in Wireless Sensor Networks is inherently faulty and unpredictable [522], due to various reasons. First, sensor nodes are likely to be quite fragile (specially in large-scale deployments), and they may fail due to depletion of batteries or destruction by an external event. In addition, nodes may capture and communicate incorrect readings because of environmental influence on their sensing components. Second, radio links are failure-prone [495], causing network partitions and dynamic changes in network topology. Links may fail when permanently or temporarily blocked by an external object or environmental conditions; mobile nodes can get out of communication range. Third, congestion may lead to packet loss. Congestion may occur due to a large number of nodes' simultaneous transition from a power-saving state to an active transmission state in response to an event.

All these fault scenarios are worsened by the multi-hop communication nature of Wireless Sensor Networks, i.e. it often takes many hops to deliver data from a sensor node to the sink; therefore, the failure of a single node or link may lead to missing reports from an entire region of the sensor network. Additionally, congestion that starts in one local area can propagate all the way to the sink and affect data delivery from other regions of the network. Fault-tolerance is thus as critical as other performance metrics such as energy efficiency or timeliness.

Fault-tolerance in WSNs must be tackled differently from more "traditional" communication networks, due to the following reasons: (1) traditional network protocols are generally not concerned with energy consumption, since wired networks are constantly powered and wireless ad hoc devices can get recharged regularly; (2) traditional network protocols aim to achieve point-to-point reliability, whereas WSNs are concerned with reliable event detection; (3) in Wireless Sensor Networks, node failures occur much more frequently than in wired, where servers, routers and client machines are assumed to operate normally most of the time; this implies improved network management mechanisms not compromising energy-efficiency and traffic overhead.

3.3.4.2 State of the Art

Common practices for robust software/algorithms are e.g.: 1) write "generic" code that can accommodate a wide range of situations and thereby avoid having to insert extra code into it just to handle special cases (code added just for special cases is often buggier than other code, and stability problems can become particularly frequent and/or severe from the interactions among several such sections of code; 2) using formal techniques, such as fuzzy testing, to test algorithms since this type of testing involves invalid or unexpected inputs/stimulus; 3) providing each application with its own memory area and prevent it from interfering with the memory areas of other applications or of the kernel. These techniques, allied with a careful resource management will lead to improved system robustness and in general to higher reliability.

There are different fault-management techniques at different layers of the protocol stack, according to the taxonomy presented in [450] for traditional distributed systems: 1) fault prevention (to avoid or prevent faults); 2) fault detection, to use different metrics to collect symptoms of possible faults; 3) fault isolation, to correlate different types of fault indications received from the network, and propose various fault hypotheses; 4) fault identification, to test each of the proposed hypotheses in order to precisely localize and identify faults; 5) fault recovery, to treat faults, i.e., reverse their adverse effects. Some proposals are built upon combinations of these.

Most fault avoidance techniques operate in the network layer, adding redundancy in routing paths; a majority of fault detection and recovery techniques operate at the transport layer; and a few fault recovery techniques perform at the application layer, concealing faults during off-line data processing. Fault prevention techniques aim to prevent faults from happening by (1) ensuring full network coverage and connectivity at the design and deployment stages as proposed in [302] [502] [209], (2) constantly monitoring network status and triggering reactive actions if deemed necessary, or (3) enforcing redundancy in the data delivery path, hoping that at least one of the paths will survive and fulfill the task of data delivery.

Monitoring the status of a Cooperating Object system, as in traditional distributed systems, provides a fundamental support for efficient network management. This can be performed either in a passive mode (observing the traffic already present in the network to infer network condition) or active mode (probes injected into the network or relying on reports from the nodes) which may present more overhead. In this line, some techniques were proposed to monitor different network parameters in Wireless Sensor Networks, like: (1) Node Status Monitoring, concerning the energy level of a node, like eScan [524] or energy map [307]; (2) Link Quality Monitoring, tracking the quality of channels at the link layer, enabling higher level protocols to adapt by changing routing structures as proposed in [495] by performing snooping; (3) Monitoring Congestion Level, by observing the buffer length as proposed in [395] and in CODA [477] .

Multipath routing has been used in traditional wired networks to provide load balancing and route redundancy: load balancing helps to balance energy consumption among sensor nodes, hence reducing the problem of power depletion of a particular set of nodes; route redundancy increases the chances of messages reaching the destination, thus improving reliability. Some proposals are GRAB [526] , Node-Disjoint Multipath [138], and Braided Multipath [138].

Nevertheless, even with fault prevention mechanisms failures will still occur, so fault detection techniques need to be in place. Here, packet loss can be used as an indicator of faults (e.g. PSFQ [476] and GARUDA [346]). Other metrics such as interruption, delay or lack of regular network activity are also considered as symptoms of faults [439] [367], as well as buffer occupancy level and channel loading conditions [395], [477].

Upon detecting alarms, fault isolation and identification will diagnose the causes. For instance, when a sink does not hear from a particular part of the routing tree, it is unknown whether it is due to failure of a key routing node, or failure of all nodes in a region. Within this concept of Fault Identification, Sympathy [367] is an interesting proposal. It determines whether the cause of failure is in node health, bad connectivity/connection, or at the sink by using an empirical decision tree.

In general, faults can be (1) discovered and recovered within the sensor network; or (2) concealed at a sink after collecting and analyzing the readings.

Faults can be recovered independently of applications, like CODA [477]. However, this type of application-independent recovery does not differentiate between important (e.g., a new report) and unimportant packets (e.g., redundant reports, control packets). On the other hand, application aware fault tolerant protocols try to achieve application specified metrics (e.g., the percentage of distinct packets delivered), which requires the nodes to analyze packets and take different actions based on packet types.

There are different proposals for ensuring reliability in data collection in upstream communications, according to the data collection mode (raw or aggregated data). ESRT [395], PERG [167] and TAG [79] are some examples. Also, for downstream communications, other techniques were already proposed in the literature. PSFQ [476], GARUDA [346], and ReACT [363] are among the most popular.

In order to provide a higher level solution for fault-tolerance, fault-management frameworks with complete management infrastructures and information models have been also proposed. Architectures like Digest [521], SNMS [456], AgletBus [273], and MANNA [390] can be complemented with previous discussed approaches to achieve better performance.

In summary, new techniques are necessary for Cooperating Object applications to be robust and reliable, in a way that other QoS properties are also respected, such as timeliness and energy-efficiency.

3.3.5 Mobility

3.3.5.1 Description and Relevance

A Cooperating Object system may involve a diverse set of entities (refer to 3.3.2). As it can be easily inferred from the application scenarios presented in this document, mobility will be a key issue in most Cooperating Object systems as they will be physically or logically moving relative to each other. Physical mobility mainly refers to the changes of the entity's geographical locations during time, such as the movement of vehicles, animals, humans. Logical mobility refers to the dynamic changes in the network topology such as adding or removing new entities in the system.

Mobility can be classified according to the type of mobile entity into three classes:

- **Node mobility:** Cooperating nodes may move isolated (single node moving) or in groups (e.g. node cluster in a vehicle). Router/gateway nodes might also be mobile for allowing to improve QoS properties (e.g. network throughput, message delay or nodes lifetime) in certain areas or to tolerate other router/gateway failures.

- **Sink Mobility:** Information sinks (maybe multiple) may be moving, either on purpose (e.g. data mules) or due to the application requirements. Machines and other equipments belonging to the Cooperating Object system can also be mobile: an autonomous vehicle, a roller belt, a crane. The Cooperating Object system may also include mobile controllers/observers or controlled/observed humans and animals.

- **Event Mobility:** It can also be considered a kind of mobility, such as in tracking applications (e.g. tsunami, gas leak, herd, fire). This mobility class must be considered, to dynamically improve QoS in the regions where events occur.

Mobility can also be classified according to mobility speed into two broad types:

- **Fast mobility:** refers to the motion of Cooperating Objects at high relative velocities (>20 km/h). An example is vehicle-to-vehicle communications for intelligent roads.

- **Slow mobility:** concerns the communication scenarios involving low relative speeds (<20 km/h), e.g. an operator with a handheld terminal moving in a factory floor.

Additionally, mobility can be classified according to mobile entities crossing or not the radio cell/cluster boundaries:

- **intra-cell (or intra-cluster) mobility:** requires no mobility management mechanism; for instance, if several users of an ad-hoc network are moving around in a room, they might not lose connectivity, so no mobility management is required;

- **inter-cell (or inter-cluster) mobility:** in infra-structured wireless networks, when a wireless/mobile node moves outside the radio coverage of a certain cell/cluster into another cell/cluster, a hand-off (or hand-over) management mechanism is required.

Note that in many application scenarios; it is not enough that the wireless protocol supports joining and leaving of nodes, since this process might lead to inadmissible network inaccessibility times, unbounded message delays or message losses for many CO applications.

Mobility support can significantly increase the capability of a Cooperating Object system, e.g.:

- Mobility can be used to maintain and repair network connectivity [69], e.g. to guarantee connectivity with sinks/controllers, leading to a self-configuration capability.
- Mobility improves network coverage [275], since initial node deployments may be extended afterwards.
- Node mobility may help to homogenize energy consumption, namely for rotating routers that are closer to information sinks.
- Node mobility can help Cooperating Object systems to adapt to dynamic stimulus changes, e.g. to collect information when a sudden incident occurs.
- Mobile sinks can improve the lifetime of WSNs by sweeping the network area and collecting data from sensors (e.g. data mules).
- Event mobility mechanisms can provide better QoS support for critical regions, which are involved in the tracking and the reporting of critical events.

The mobility concept in Cooperating Object systems is therefore rather heterogeneous and challenging.

3.3.5.2 State of the Art

In most of the works related to WSNs, topology dynamics results mainly from sensor nodes failure rather than from the mobility of nodes (sensors or sinks). In other words, sensor nodes (and the physical topology) are assumed to be static during runtime. While this assumption might be true for certain applications, it completely fails for other types of applications such as health care monitoring [426], disaster emergency response [81], monitoring hostile environments [194], tactical operations (e.g. airport control, home security, military operations) aiming to track any target that enters the field [57].

While some existing wireless communication protocols support inter-cell mobility (WiFi, GSM), mobility support in Wireless Sensor Networks is still in its infancy. Even if most

WSN protocols support joining and leaving of nodes (e.g. ZigBee), mechanisms that enable transparent, energy-efficient and reliable mobility without network inaccessibility times are still missing (e.g. [321], [253], [447]). Moreover, fast mobility turns the meeting of these requirements even harder.

Mobility models are generally described with stochastic models taking into account mobility parameters such as speed, movement direction, radio propagation models and presence of obstacles. Examples of mobility models includes Random Waypoint Model (RWM), Random Walk Model (RWM), Brownian Motions, Gauss-Markov (GM), City Section Mobility (CSM) [55]

Mobility support greatly impacts Cooperating Object system design, namely in what concerns lower protocol layers and particularly MAC and routing mechanisms.

Mobility influences MAC protocols design for two reasons. First, mobility involves topological changes that may affect algorithms that need to tune some parameters according to the density of nodes in the contention area (SIFT, TRAMA, TSMA, MACAW) . Second, MAC algorithms based on medium reservation (MACA, MACAW) may fail in case of mobility, since the reservation procedures usually assume static nodes. For instance, algorithms based on the RTS/CTS handshake to reserve the medium may fail because either the corresponding nodes move outside the mutual coverage range after the handshake or external nodes get into the contention area and start transmitting without being aware of the medium reservation. Nevertheless, many MAC algorithms can self-adapt to topology variations in case of nodes mobility (TRAMA, TSMA and SMACS-EAR) , but at the expense of higher energy consumption and medium access delay.

Generally speaking, many routing algorithms are able to cope with topology dynamics resulting from nodes mobility. However, most of them react to topology variations by dropping the broken paths and computing new ones from scratch, thus incurring in performance degradation. In particular, mobility may strongly affect cluster-based algorithms, due to the high cost of maintaining the cluster-architecture over a set of mobile nodes. Some routing algorithms specifically designed for networks with slow mobile nodes (e.g. GAF, TTDD) attempt to estimate the nodes trajectories. The SPIN family of protocols is well-suited for environments where the sensors are mobile, since forwarding decisions are based on local neighborhood information.

3.3.6 Security

3.3.6.1 Description and Relevance

Cooperative Objects have raised keen interest of the research community in coordination problems, communication protocols, and algorithm distribution. Such interest has lead to the spreading realization of such developments in real applications such as industrial and building automation, military surveillance and so forth. However, given the interactive and pervasive nature of Cooperating Objects, security is one of the key points for their acceptance outside the research community. In fact, a security breach in such systems can result in severe privacy violations and physical side effects, including property damage, injury and even death.

Security in Cooperating Objects is a more difficult long-term problem than is today in desktop and enterprise computing. In fact, such objects that are in spatial proximity cooperate among themselves in order to jointly execute a given task. It follows that there is no central, trusted authority that mediates interaction among them. Furthermore, Cooperating Objects often use wireless communication in order to simplify deployment and increase reconfigurability. So, unlike a traditional network, an adversary with a simple radio receiver/transmitter can easily eavesdrop as well as inject/modify packets in a wireless network.

Cost reasons often cause to have devices with different on-board sensors and heterogeneous in terms of energy, computation, and communication capabilities. This leads to constraints on the types of security solutions that can be applied. To further worsen this scenario, devices often lack adequate physical/hardware support to protection and tamper-resistance. This, together with the fact that Cooperating Objects can be deployed over a large, unattended, possibly hostile area, implies that each device can be enforced in different ways by careless, or even malicious administrators.

Finally, the drive to provide richer functionality, increased customizability and flexible reconfigurability of Cooperating Objects requires the ability to dynamically download software on them [372] [371]. In fact, traditional systems have been designed to perform a fixed set of predefined functionalities in a well-known operating environment. Hence, their functionality is not expected to change during the system lifetime. This design approach can no longer be pursued in the vast majority of applications. In order to be cost-effective and operational over time, Cooperating Objectsare required to be reconfigurable in order to be customizable to different operating environments and adaptable to changing operating conditions. However, the need for reconfigurability acts against security as it introduces new sources of vulnerability. Downloading malicious software (including viruses, worms, and Trojan horses) is by far the instrument of choice in launching security logical attacks. The magnitude of this problem will only worsen with the rapid increase in the software content of embedded systems.

3.3.6.2 State of the Art

In such scenario, we focus on the following security breaches that have to be address in order to guarantee the safety and reliability of the overall system:

1. **Security bootstrapping:** Security services are essential to ensure the authenticity, confidentiality, freshness, and integrity of the critical information collected and processed by Cooperating Objects. An open research problem is how to bootstrap secure communications among devices, i.e. how to set up secret keys among communicating

nodes? This key agreement problem has been widely studied in general network environments. There are three types of general key agreement schemes: trusted-server scheme, self-enforcing scheme, and key pre-distribution scheme. The trusted-server scheme depends on a trusted server for key agreement between nodes, e.g., Kerberos [330]. This type of scheme is not suitable for networks without a trusted infrastructure. The self-enforcing scheme depends on asymmetric cryptography, such as key agreement using public key certificates. However, limited computation and energy resources of devices often make it undesirable to use public key algorithms, such as Diffie-Hellman key agreement [98] or RSA [378]. The third type of key agreement scheme is key pre-distribution, where key information is distributed among all sensor nodes prior to deployment. If we know which nodes are more likely to stay in the same neighborhood before deployment, keys can be decided a priori. However, because of the randomness of the deployment, knowing the set of neighbors deterministically might not be feasible. In such case, the research community proposed key management schemes aimed at establishing secure communications among devices from a collection of pre-deployed keys without any previous knowledge of the network topology [530] [121].

2. **Key management: distribution and revocation:** Faulty or malicious devices have to be logically removed from the network communication in order to guarantee the system availability and safety. Usually, the ability to logically remove compromised devices from the network translates into the ability to revoke keys [66]. In fact, cryptographic algorithms do not expose keys so that secret keys can only be compromised by compromising the device. It follows that by revoking all keys of a compromised device, it is possible to remove the logical presence of that device from the system. Several group key management systems have been proposed so far [75], [117], [480], [147], [529]. Some of them suggest grouping strategies aimed at reducing the overhead of group key management [117], [147]. Other systems instead group neighboring sensor nodes and iteratively merge groups up to establish network-wide shared key [75], [530], [101]. Approaches proposed in as [117], [147], [480] are centralized in contrast to distributed schemes proposed in [75], [530]. In a centralized scheme, the key distribution center constitutes a single-point-of-failure and may cause a performance bottleneck. However, many distributed approaches are static and thus falls short in supporting rekeying and node revocation.

3. **Secure Reconfiguration:** Cooperating Objects typically are subject to unpredictable changes of operational conditions so that they have to be able to self-reconfigure in order to meet the changing conditions. For instance, as soon as an emergency situation occurs, the sensor nodes may need to change their task due to the changed operational conditions. Reconfiguration may concern the node task as well as the implementation of a given service. As a further example, a mobile agent with on board sensors for self-localization, such as cameras, may need to localize

itself differently in case of drastic light changes. Mechanisms for secure reconfiguration comprise remote downloading of authenticated components into a device after it has been deployed. In case the remote downloading takes place through the wireless network, an attacker may modify a component in transit or install a rogue one. Therefore, it is vital to authenticate the source of a component as well as verify its integrity. In addition, authenticated downloading must be efficient in terms of communication, storage and computation in order to mitigate potential denial of service attacks against resource poor devices. Specific security threats, vulnerabilities and related countermeasures have been extensively discussed in [235], [372], for example. An FPGA-based reconfigurable computer architecture layer has been proposed in [153], whereas [110] defines a scheme for authenticated downloading of software that has been conceived for low-end resource-poor devices.

4. **Intrusion Detection System:** An Intrusion Detection System (IDS) has to be able to detect a third party's attempts of exploiting possible insecurities and warn for malicious attacks, even if these attacks have not been experienced before. Intrusion detection is an important aspect within the broader area of computer security so that there is currently a keen interest of the research community in this area. Extensive work has been done in IDS for cooperative networks such as ad-hoc networks [279], [34], [308], [223]. There are three main techniques that an IDS can use to classify actions [21]: misuse detection, anomaly detection and specification-based detection. In misuse detection or signature-based detection systems, the observed behavior is compared with known attack patterns (signatures). Action patterns that may pose a security threat must be defined and stored to the system. Then, the misuse detection system tries to recognize any "bad" behavior according to these patterns. It is worthwhile to notice that ad-hoc networks with severe memory constraints make signature-based detection systems relatively difficult to build and less likely to be effective [308]. Anomaly detection systems focus on normal behaviors, rather than attack behaviors. First these systems describe what constitutes a "normal" behavior (usually established by automated training) and then flag as intrusion attempts any activities that differ from this behavior by a statistically significant amount. Finally, specification-based detection systems are also based on deviations from normal behavior in order to detect attacks, but they are based on manually defined specification that describe what a correct operation is and monitor any behavior with respect to these constraints. In particular, the authors in [279], [34] describe specific IDS for routing attacks in ad-hoc networks. In fact, the peculiar characteristics of these routing protocols are used in order to detect misbehaving devices. In [88] [340], the authors propose similar IDS systems, based on a certain number of nodes responsible for monitoring their neighbors and looking for intruders.

5. **Secure routing:** Cooperative objects actively communicate among each other in order to perform their own tasks. Such objects usually perform routing protocols to

forward data packets from the source towards the destination. Without any secure mechanisms a malicious or faulty devices can perform any actions in the packets it forwards. It is worthwhile to notice that security at routing level is very important, because if the routing is compromised, other protocol layers on top of the network layer are also compromised. An attack at routing level falls into one of the following categories: i) spoofed, altered, or replayed routing information, ii) selective forwarding, iii) sinkhole attacks, iv) wormholes, v) acknowledgment spoofing [102], [223], [529]. Some of the attacks can be performed by dropping, changing, or injecting packets into the network. Other attacks are executed by changing the real topology of the network. In order to guarantee secure routing a number of approaches have been designed for different environments and security objectives. Usually certificates, digital signatures or HMACs provide in principle authentication and integrity of messages. Depending on the network assumptions and requirements a routing protocol hinges on asymmetric cryptography, symmetric cryptography, or reputation systems. In particular, ARAN [399] hinges on asymmetric cryptography so that each device has a public/private key pair that bound with additional information is used as a certificate. In ARIADNE [197] , each pair of devices shares a secret symmetric key or a nonce. The shared keys are used to generate keyed-hash message authentication codes, while the nonce is used by one-way hash functions in order to generate hash chains, or hash tree chains. Some protocols adopt more than a single mechanism (e.g., SAODV [516] and SEAD [196]) in order to secure the network at routing level. However, cryptographic schemes are defenseless against attacks from compromised nodes. Thus, reputation systems can be used complementary with cryptography to better achieve security against both malicious and faulty devices. In particular, the protocols CONFIDANT [47] and WatchDog-Pathrater [299] adopt this approach. Another method for defending against compromised nodes is the usage of mechanisms that perform plausibility checks over the received data. This latest mechanism is adopted by DCMD [154] avoiding the scalability and mobility problems, that are in reputation systems

3.3.7 Heterogeneity

3.3.7.1 Description and Relevance

Cooperating Objects systems will inherently be composed of heterogeneous components, therefore heterogeneity must be appropriately considered both pre-runtime (at design time) and during system operation (e.g. for system management and maintenance).

In our context, we consider heterogeneity in a broad perspective and at different levels:

- heterogeneity in networking hardware/software
  - sensor/actuator-level nodes (motes, RFID)

  - sensor/actuator-level communication protocols
  - higher-level nodes (e.g. gateways)
  - higher-level communication protocols
  - network planning/management

- heterogeneity in embedded system nodes hardware/software architecture
  - sensors and sensor boards, design diversity, calibration
  - operating systems for resource-constrained networked embedded systems
  - programming languages for resource-constrained networked embedded systems
  - middleware

- heterogeneity in cyber/pervasive/host computing devices
  - HMIs (in general)
  - wearable computing (e.g. PDAs, HMDs)
  - mobile robots, transportation vehicles
  - machinery

- heterogeneity in applications/services/user-perspective
  - many applications/services may be provided; same networking infrastructure
  - potentially many different human users, eventually playing at different levels

The integration of heterogeneous objects featuring different embedded information processing and communication capabilities has a huge number of application possibilities. Furthermore, Cooperating Object systems featuring heterogeneous hardware offer the additional advantage of exploiting the complementarity and specialization of each object. Nevertheless, it must be highlighted that system design/management complexity grows (even more that linearly) with heterogeneity.

3.3.7.2 State of the Art

Concerning networking hardware/software, it is commonly accepted that the integration of Radio-frequency Identifier (RFID) technology with Wireless Sensor Networks provides a symbiotic solution that leads to improved performance of the system. Actually, there is a growing convergence between WSN nodes and RFID nodes technology, particularly for the case of RFID devices with active characteristics, where both computation and communications modules are battery-powered. In this line, the frontier between the two types of technologies is getting increasingly undefined.

Sensor/actuator-level communication protocols might also be different, due to different factors, such the Cooperating Object system covering several geographical locations with different communication requirements (e.g. waiting room vs. chirurgic room in a hospital) or the Cooperating Object system being composed of several subsystems that have gradually been added over time (e.g. upgrades, extensions). Gateway-like devices might need to guarantee the interoperability with heterogeneous higher-level communication protocols (e.g. WiFI or WiMAX). Network planning/management tools must tackle these heterogeneous systems in an efficient and straightforward fashion, which is certainly a big challenge.

Heterogeneity also arises in terms of embedded system nodes hardware/software architecture. Different types of sensors and sensor boards may be used for measuring different physical parameters, which brings additional complexity to the Cooperating Object system, e.g. in what concerns calibration. "Design diversity", i.e. using heterogeneous components to perform the same task (e.g. measuring the same physical parameter with two different types of sensors or performing the same computation using two different processing units), might also be required in more critical applications. Different operating systems and programming languages (particularly for resource-constrained networked embedded systems) might also be required. For example, wireless sensor/actuator nodes in a certain Cooperating Object system might run different operating systems. Also middleware such as for the provision of security or fault tolerance might be quite heterogeneous.

Different types of hosting/client equipment may be simultaneously be used in a Cooperating Object system. Wearable computing equipment (e.g. mobile phones, handheld terminals, PDAs, HMDs, RFID readers), database servers, other HMI-computers, mobile robots, transportation vehicles or industrial machinery are just a few examples illustrating the high level of heterogeneity that can emerge in Cooperating Object systems.

It is also accepted that the underlying (most of them large-scale) networking infrastructures of Cooperating Object systems will likely support many applications and services, most probably each of them imposing different QoS requirements, e.g. strongly depending on spatiotemporal status. Imagine a Cooperating Object system for building automation. It may control security/access control, fire/smoke alarm systems, HVAC system, lights, doors, blinds, lifts/escalators, each of these with particular requirements that will very dynamically depend on space (location) and time (evolution). Also, Cooperating Object systems will most probably involve many different human users, playing at different levels, with different cultures and technical skills, bringing important challenges to Cooperating Object system designers. In a vertical systems' perspective, objects mobility also brings additional complexity to system design and management. Mobility capabilities can exist at different system levels, ranging from mobility of a single sensor/actuator node, of groups of sensor/actuator level nodes (e.g. node cluster in a human body), whole wireless sensor/actuator networks, gateways or mobile robots. The efficient (e.g. time, energy, reliable, scalable) cooperation between heterogeneous static and mobile objects is thus of high complexity.

3.4 Systems

In this chapter, different approaches to build complete Cooperating Object systems are evaluated. While pure operating systems provide basic support for application developers more advanced middleware solutions already include special functionality and abstractions. Finally, the integration of Cooperating Objects in bigger systems is described.

3.4.1 Operating Systems

Since Cooperating Objects comprise very heterogeneous platforms a multitude of operating systems exist that are tailored to these platforms. Nevertheless, the main purpose of all operating systems is at least the provision of a hardware abstraction and the management of resources. Depending on the target platforms, a multitude of other functionality might be included.

If PCs are part of a Cooperating Object like in Virtual Environments standard operating systems such as **Windows** and **Linux** are used. For smaller devices with GUI like PDAs or embedded systems like routers or music players there exist related or scaled down versions of these systems. **Windows CE** is similar to Windows but based on a different kernel. Using Microsoft Platform Builder, only the necessary modules can be selected and deployed on the devices. **uClinux** can be used in systems without a memory mapping unit and thus no isolation between kernel and user-space processes, **Mobilinux** is specially adapted to ARM based mobile phones. Since Linux is covered under the GPL license a multitude of other scaled-down versions exist that would go beyond the scope of this section.

More operating systems are specially designed for handheld devices such as PDAs and mobile phones: **Palm OS Cobalt** and **Symbian OS** both feature multi-tasking, memory protection and a multimedia and graphic framework. Both run exclusively on ARM processors.

Beside operating systems for mobile phones that are based on real-time kernels more embedded real-time systems exist. A few of them are presented here: **VxWorks**, a commercial and proprietary system, is able to run on practically all current CPUs used for embedded systems. It features preemptive multitasking, memory protection, different communication and synchronization mechanisms, and error management. The smallest default configuration has a footprint between 36 and 100 KBytes. VxWorks is used in cars, aircrafts, and spacecrafts. **FreeRTOS** is a scalable open-source system which is designed to be “small, simple and easy to use”. It features different scheduler operations, different communication and synchronization mechanisms, and is free to use in commercial applications. **eCos** [113] is an open-source system that has extensive configuration possibilities and can be scaled up from a few hundred bytes in size to hundreds of KBytes. It provides features such as preemptable tasks with multiple priority levels, low latency-interrupt handling, multiple scheduling policies, and multiple synchronization methods. **QNX** [360] is a Unix-like micro-kernel system enriched with cooperating processes that provide higher level services such as inter-process and low-level networking communication, process scheduling and interrupt dispatching. It features a very small kernel of about 12 KBytes.

In the area of sensor networks, many operating systems have been developed. **TinyOS** [186] is the oldest and most known representative of this class. It is not only adapted to very resource constraint processors, for example 8-bit RISC from Atmel or Texas Instruments, but also tailored to complete sensor network platforms that have additional external devices like a radio module or flash memory and external sensors. TinyOS builds on a component architecture where both applications and operating system consist of single, interlinked components. NesC, an extension to the C language, allows to wire these components. During compilation, the operating system and the application components are combined to a single program. TinyOS uses a two level scheduling hierarchy that lets high-priority events preempt low priority tasks. Events are invoked because of external input such as incoming data, sensor input or a timer. Both events and tasks run to completion after being started. In this event-driven concurrency model blocking calls are not permitted and, therefore, application is split into several parts where a command initiates an action and an event handler processes the results after the action has been completed.

This programming model requires the application logic to be distributed over several functions, thus becoming hard to understand and maintain. Several approaches try to improve this by introducing lightweight, thread-like abstractions. **Fiber** [485] introduces a single long-running blocking execution context in which the application can run with only 24 bytes of RAM and 150 instructions of runtime overhead on AVR ATmega processors. **TinyThread** [301] is a library for cooperative multi-threading where each thread has its own stack. The size of each stack is fixed, but computed automatically in an effective and accurate way using a stack-estimator tool. The cost for context switching is 168 instructions on AVR ATmega processors and 33 instructions on MSP430 processors. textbfY-Threads [336] are preemptive multi-threads with small stacks since the majority of work are "run to completion routines" that execute on a common separate stack. On ATmega processors, "run to completion routines" have an overhead of 83 instructions to normal subroutine invocations and lightweight threads have an overhead of 368 instructions.

Other operating systems for networked and memory constrained systems tried to avoid the TinyOS deficiencies by design. **Contiki** [105] supports several execution models, dynamically loadable programs. Its kernel is event-driven, multi-threading is implemented as optional library. Additionally, Contiki provides **ProtoThreads** [108], stack-less thread-like constructs that are extremely lightweight requiring only two bytes of memory per ProtoThread and no additional stack. One of the recently added Contiki features is the Coffee flash-based file system. Coffee makes it possible to store data as files on flash-based memories such as on the on-board flash ROM on the TelosB/Tmote Sky. Developers can use Coffee through the Contiki file system interface. The Contiki shell simplifies use and maintenance of deployed sensor networks by supporting network-level commands, low-power radio networking, sensor data collection, and power profiling. In October 2008 Cisco, Atmel, and SICS announced uIPv6, the world's smallest open source compliant IPv6 stack, for Contiki. uIPv6 passes all the tests required for an IPv6 stack to be called IPv6 Ready. **Mantis OS** [33] features preemptive priority-based multi-threading, reprogramming and debugging over the air, while having a footprint of less than 500 Bytes RAM and 14 KBytes ROM. **LiteOS** [58] is a multi-threaded operating system that supports dynamic loading and online debugging and provides wireless Unix-like shell interface and a hierarchical file system, thus simplifying the programming and handling of sensor systems. LiteOS is designed for MicaZ and IRIS node; the size of compiled programs is comparable to TinyOS.

Some operating systems, e.g. SOS, kOS, Timber or DCOS, have been developed with specific design goals like dynamically-loading modules, suitability for iterative applications, individual tailoring to applications or data-centric architecture, but they are not maintained any more.

3.4.2 Middleware

A general goal of a middleware is to hide the complexity of the underlying platforms or infrastructure. In the area of Cooperating Objects, very different approaches exist to achieve this goal. A middleware can hide the complete platform or the distributed nature of the network from the user by providing high-level programming abstractions with Macroprogramming or Virtual Machines. On the other hand, a middleware can support the application programmer by providing additional functionality to the pure operating system, e.g., resource naming, distribution of tasks, adaptive behavior, data sharing, service invocation, event detection and context management.

3.4.2.1 Macroprogramming

The idea of macroprogramming approaches is to view and program the network as a whole using a high-level abstraction and a compiler generates node-level code from it. This eases the development significantly but leaves room for improvement if node-level code is written by experienced programmers.

In **Regiment** [331], the network appears as a set of spatially-distributed time-varying signals that are either raw sensor readings, computations thereof or aggregation of regions. A region is a collection of signals defined by spatial, topological or logical expressions. The user can modify, filter or aggregate the streams in a region, thus forming new regions or an aggregate signal. Regiment programs are first compiled to 'token machines' in nesC and then treated as normal TinyOS programs. **Kairos** [160] is a preprocessor add-on to the native language. It provides three abstractions: nodes, which are named using integer identifiers, and lists thereof, one-hop neighbors, and remote data access on the named nodes, which is a shared-memory abstraction across nodes with an 'eventual consistency' model. Currently, only a Python implementation of the Kairos primitives exists. **Flask** [287] provides functional programming for sensor networks. It is a domain specific language embedded in Haskell, but separates between node-level code and the meta-language used to generate it. It allows also to include nesC in Haskell code to have access to existing sensor network code. Flask mainly targets streaming data applications.

3.4.2.2 Virtual Machines

A virtual machine offers to applications a suite of virtual instructions and maps them to the real instruction set actually provided by the underlying real machine. In this way, the virtual machine abstraction can mask differences in the hardware and software lying below the virtual machine itself, thus facilitating code and data mobility. Since the virtual machine code can be made smaller the energy consumption of transmitting the code over the network can be reduced. Depending on the capabilities of the underlying devices, the virtual machine in turn can offer more powerful functionality.

Java Platform, Micro Edition (Java ME) comes in two configurations, one for mobile phones and small PDAs (CDLC - Connected Limited Device Configuration) and one for more capable devices like smart phones, set top boxes and embedded devices (CDC - Connected Device Configuration). Both configuration can be extended with profiles specifying a set of higher-level APIs for specific devices. For example, the Mobile Information Device Profile for CDLC on mobile phones provides a better user interface, game and media support, but also the ability to dynamically deploy and update applications wirelessly. Additionally, optional packages exist with technology-specific APIs. Code migration is possible from Java SE to CDC or CLDC to CDC (without profiles). **Sentilla Point** is a commercial pervasive runtime environment, which is CDLC compliant and additionally features networking, sensing, a file system and energy management. It is integrated in a software solution to develop, install and debug programs for these platforms as well as to connect the pervasive computers to the enterprise system. The **Microsoft .NET Compact Framework** is intended for Windows CE based devices like mobile phones, PDAs or set top boxes. It shares some common libraries with the full .NET framework but includes also some platform specific libraries. In contrast, the **.NET Micro Framework** is even more restricted to fit very small devices with a memory footprint of only 300 KBytes. Code is interpreted and not compiled using the Just-In-Time compiler like in the Compact Framework and only C# is supported.

In contrast to the Java and .NET approaches, the dynamic nature of Cooperating Objects is reflected by Virtual Machines that allow reconfiguration and mobile code. **SensorWare** [42] targets medium-sized Cooperating Objects like iPAQs. Programs are written in Tcl, and SensorWare specific functionality is implemented as a set of additional procedures, e.g. querying of a sensor, sending of messages or waiting for events. Tcl scripts can be sent to and installed on all or specific neighbors using a special command. **VM*** [240] is a software framework for building virtual machines in heterogeneous environments. Its key observation is that a virtual machine running on a device does not have to support the full VM specification but only the services needed by the running application. Therefore, VM* is able to update both the virtual machine and the application on a node incrementally.

The current implementation includes a component-based Java Virtual Machine, but the concept is more general. Since every node might have different capabilities, code updates are controlled by a computer running the VM* framework in contrast to the viral code dissemination in the other approaches.

Maté. [268] is a byte-code interpreter for very resource-constrained platforms that allows to concisely describe a wide range of sensor network applications through a small set of common high-level primitives. Therefore, complex programs can be implemented very short. Eight instructions can be defined by the user in a tailored version of Maté. The system knows three execution contexts, clock timers, message receptions and message send requests, that can run concurrently. A program can broadcast itself to the neighbors of the current node, thus disseminating through the network. **ASVM** [269] (Application Specific Virtual Machines) is an enhancement of Maté and addresses its main limitations: ASVM supports a wide range of application domains, whereas Maté is designed for a single domain only. On system events, handler trigger threads that are executed in a FIFO round-robin model, while a concurrency manager ensures race-free and deadlock-free execution. Code propagation is not only done via broadcasts, but with a control algorithm based on Trickle to detect when code updates are really needed on other nodes. **Melete** [513] further improves on ASVM to support concurrent applications on a single node and dynamic grouping of sensor nodes for group-keyed code dissemination above Trickle.

3.4.2.3 Network Level Abstraction

Like for macroprogramming approaches, the abstraction of Virtual Machines also have the power to hide the distribution aspect of Cooperating Objects. **SINA** [422] models the network as a collection of massively distributed objects. The user interacts with SINA using a procedural scripting language called SQTL (Sensor Querying and Tasking Language) that features primitives for hardware access, location-awareness, communication and event handling, but also an SQL-like declarative query language. SQTL scripts are injected from a front-end node into the network and the script decides if it pushed itself to other nodes to accomplish its task. Data is sent explicitly to other nodes, e.g., the neighborhood or the node the script was received from. **MagnetOS** [276] make the entire network appear as a single Java virtual machine. Regular Java applications are rewritten at byte-code level by a static partitioning service into distributable components that communicate via events. Several algorithms in the core of the operating system decide when and where to move application components including all their state, trying to shorten the mean path length of data packets sent between components of an application by moving communicating objects to topologically closer nodes.

3.4.2.4 Resource Naming

If Cooperating Objects are spread in the physical world access to data and services needs network-transparent abstractions. In general, locations are combined with a description of the resource needed to form a spatial reference. The networked embedded system is seen as a single virtual address space. Interestingly, all of the following systems are extensions to Java and make use of Smart Messages that consist of code, data and execution state and can migrate to nodes of interest to execute the contained program there. In **Spatial Programming** [39], a complete spatial reference has the form "{space:tag[index]}.resource": space defines a physical region, tag a property or service, index distinguishes multiple systems since tag is not unique, and resource selects the specified resource from the selected system. **SpatialViews** [333] are virtual networks consisting of nodes named by their services and locations. A spatial iterator discovers all matching nodes in a SpatialView and migrates computation to them, which can also occur in parallel by replication. SpatialViews can have a space granularity and iterators a time granularity, i.e. a node can be revisited after the given time interval or if it has moved more than the given distance. **Declarative Resource Naming** [207] allows to specify resources using a boolean expression that can include space, services, but also properties of a node (e.g., sensor readings) and user defined functions. Matching nodes can be accessed sequential or parallel. In the latter, network aggregation can be performed, as well.

3.4.2.5 Task distribution

The overall task of a network of Cooperating Objects can usually be divided into several subtasks. An important and interesting problem is to distribute these subtasks to single devices according to their resources and capabilities.

In **SORA** [288] (Self-Organizing Resource Allocation) sensor nodes are seen as agents that perform actions to produce goods (e.g., sensor readings or data aggregates) in return for (virtual) payments while respecting their energy constraints. Goods prices are globally-advertised throughout the network and single nodes decide to perform only those action that maximize their (local) utility function, whose value depends on both the node's internal state and the payment the node will (virtually) get to perform those actions.

When nodes take on specific functions they perform a certain role in the network. Such roles may be based on varying node properties (e.g., available hardware and their characteristic, location, network neighbors) and may be used to support applications requiring heterogeneous node functionality (e.g., clustering, data aggregation). The idea of generic role assignment is to generalize the concept of self-configuration of wireless sensor nodes into a generic framework that allows to assign different roles to wireless sensor nodes without requiring manual intervention [385]. With **Generic Role Assignment** [132], a developer can specify user-defined roles and rules for their assignment using a high-level configuration language. Rules are Boolean expressions that may contain predicates over the local properties of a node and predicates over the properties of well-defined sets of nodes in the neighborhood of a node. This approach is very generic since it is also suitable for very different domains, e.g. coverage problems, clustering techniques, or the formation in-network aggregation trees. One critical factor of generic role assignment is the convergence of the role assignment process. For non-trivial rule sets, it cannot be guaranteed in general that the sensor network converges to stable roles for all nodes. Frank and Römer propose to investigate rule sets in simulations to determine whether a fixpoint configuration exists and the role assignment algorithm is likely to terminate. Moreover, they argue that the initial role values of the nodes play an essential role in the timely convergence and propose heuristic approaches for initializing the nodes with reasonable roles.

Weis et al. [484] present a role assignment mechanism that consist of an algorithm stack. On top of a general radio interface, a spanning tree is constructed, which in turn is used by a publish/subscribe system that is also able to deliver a published message to just one subscriber. In the role assignment algorithm, all nodes decide which roles of an application they can play according to their capabilities and subscribe to a message channel indicating this possibility. The root node of the spanning tree assigns each necessary role to one offering node using the special publish mechanism.

3.4.2.6 Adaptive Systems

Since the requirements to Cooperating Objects or the system environment can change significantly during the lifetime and a constant manual adjustment is too costly, several systems have been developed that perform automatic adaptation. **MiLan** [182] allows sensor network applications to specify their quality needs for sensor data based on different application state. The system monitors the availability and quality of the single sensors, the energy level of the nodes, and channel bandwidth and proactively determines which sensors need to send data and which role each sensor should play in the overall scheme. **Impala** [277] goes beyond that and supports on-the-fly application adaptation based on parameters and device failures which allows to improve the performance, reliability and energy-efficiency of the system. The adaptation capability is static since it is based on a finite state machine where different protocols are assigned to different states and conditions on the parameters represent the transitions. Impala also includes an application updater that receives, transmits and installs program updates wirelessly. **TinyCubus** [298] is a more general approach that aims at the creation of a generic reconfigurable framework for sensor networks. The Data Management Framework of TinyCubus provides a set of data management and system components. Each component is classified according to its suitability to several parameters. The framework is then responsible for the selection of the appropriate implementation based on current parameters contained in the system. TinyCubus also includes a Tiny Cross-Layer Framework which provides a generic interface to support the parameterization of components that use cross-layer interactions and a Tiny Configuration Engine which distributes and installs new code in the network.

Pervasive Computing middleware is geared towards more powerful and heterogeneous devices with very different capabilities. A distributed application running on these devices needs a coordinating instance that handles, for example, task distribution and data sharing. **Gaia** [380] uses a model-view-controller pattern and extends it with a model adapter that transforms the type of data between the model and the view. Therefore, it allows to use a varying number of input and output components by mapping the application to the available devices and services. Gaia adapts to a changing environment by changing this mapping. In **PCOM** [30], applications consist of components, whose dependencies are specified explicitly. The middleware tries to satisfy the dependencies by other components recursively, thus creating a component tree. Different components of the same application can be distributed over multiple devices. When a devices becomes unavailable or deteriorates its service or if a new device with better services becomes available PCOM can replace components to adapt to the new situation. The system of **Paluska et al.** [344] relies on "goals" and "techniques". A goal describes what functionality is needed and what properties a technique must have to satisfy it. A technique consists of sub-goals declarations, which in turn are evaluated recursively, evaluation code to compute the properties of the technique and commit code to start, update and stop application components. This allows for more flexible configuration and adaptation.

While the former systems try to adapt a single application by influencing its internal behavior, another class of adaptive middleware aims at coordinating the execution of multiple applications. Although the applications themselves are usually not distributed, different applications can run on different devices. A common example for this kind of adaptation are Smart Environments. **Aura** [139] works with a task abstraction, which is a user task, e.g. the preparation of a presentation, including its state. The task manager captures the user intents, searches for suitable services in each environment, monitors the execution and warns or reconfigures when QoS requirements are not met. The task manager can also migrate a task to another environment by check pointing the state of running services in the old environment and find and configure the services in the new one. **iROS** [358] targets interactive workspaces. Devices are independent entities that communicate via an EventHeap, which is based on the tuple space model, but extended with timeouts for tuples. Application can post notifications to the event heap, others can react to them and change their behavior.

3.4.2.7 Cross-Layer Interactions

In order to perform resource adaption, it is necessary that information can be shared across layers. For example, the application might need measurement results from the link layer to adapt its resource usage to the current quality of the wireless links.

There are many approaches in the wireless networking community where researchers have used cross-layer techniques to achieve performance improvements. For example, Van Hoesel et al. have prolonged lifetime of Wireless Sensor Networks by tightly integrating medium access and routing [188]. Cui et al. have proposed a joint optimization for link layer, MAC and routing [85]. Chiang has jointly optimized power and congestion control [72]. Song and Hatzinakos have presented a cross-layer approach that targets a specific application, namely target tracking [434]. However, with cross-layer designs replacing cleanly layered solutions there is an increasing risk of catching unwanted interactions between layers [230].

The Chameleon architecture [106], used in the Contiki operating system, introduces the concept of packet attributes to solve the problem of cross-layer information sharing while maintaining the separation of concerns from traditional layered architectures. Packet attributes are attached to both outgoing and incoming packets and allow information to flow between the layers, without any cross-layer interaction. The packet attributes hold information about physical attributes such as the signal strength for the incoming packet as well as information from the packet headers, such as sender and receiver addresses. This information is accessible through the entire stack, up to and including the application layer. Packet attributes makes information from the entire stack available to all layers, without the need for any explicit cross-layer knowledge.

Despite these improvements, there is not much architectural support for sharing information across layers. Most of the approaches presented above are implemented without architectural support from the operating system or another abstraction layer. The reason for the absence of architectural support might be that traditionally, protocol implementations have used the concept of layering. Layering separates different concerns in a network architecture in order to reduce the design complexity. The high modularity of layering, however, restricts the collaboration of different layers that could potentially benefit from sharing each others unique information.

Among the few approaches for sharing information across layers are those presented by Köpke et al. [237] and Lachenmann et al. [258]. Köpke et al. [237] identified the need for cross-layer interactions. They present a mechanism for component interactions, present the design and describe the properties of an implementation that is based on publish/subscribe. Lachenmann et al. [258] have presented TinyXXL, a language and framework for supporting cross-layer interactions. The framework provides a state repository that stores state information and configuration. It supports cross-layer interactions and reconfigurations with a publish and subscribe mechanism. Their experiments demonstrate a very low overhead in terms of memory consumption and only a little runtime overhead. TinyXXL is an extension of nesC that requires recompilation when the parameter set or the modules using the state change.

TinyOS and Contiki [105] currently do not offer standard knobs for resource tuning and parameter (re)configuration. The Pixie operating system has support for resource aware programming [280]. Pixie uses an abstraction called resource tickets similar to the ticket abstraction in lottery scheduling [475]. A tickets represents the right to use certain resources until an expiry time. Application can use resource brokers to manage resources on their behalf but they can also perform allocations themselves. This way, applications can adapt their resource consumption but finer-grained adaptation as provided by Lachenmann's solution does not seem to be supported.

3.4.2.8 Data Sharing

Devices that need to cooperate to accomplish a global task, need also to share data and information about their internal states. There are several concepts used to accomplish this task: publish/subscribe services, global and local shared information spaces with different notions of locality.

The publish/subscribe paradigm is used since many years in distributed systems. **MIRES** [436] is build on top of TinyOS and provides a publish/subscribe service for sensor networks and, therefore, suitable for resource-constraint Cooperating Objects. Using MIRES, nodes can advertise available topics. A user application at the sink node receives these advertisements and subscribes to the desired topics. After the subscriptions have been disseminated in the network nodes start to publish the data of interest. MIRES goes beyond a pure publish/subscribe service by allowing to intercept data, e.g., for aggregation.

A different abstraction is provided by shared tuple spaces. A tuple space is a multiset of tuples that are, in turn, a list of typed parameters representing the actual data. Tuples can be added to, read and removed from the tuple space. Linda introduced this model and provided a persistent and globally accessible tuple space. **LIME** [354] follows this paradigm but creates a transiently shared tuple space. Each node has access to a local interface tuple space using the same operations as in Linda. The content of this tuple space is continuously recomputed as the content of the tuple spaces of all currently connected nodes, thus providing all nodes with the same view on the tuple space. LIME extends tuple spaces with reaction methods that are executed when a tuple matches a specific pattern. LIME is implemented in Java and aims at powerful Cooperating Objects since the jar file is approximately 100 KBytes in size. **TinyLIME** [87] integrates more resource constrained devices like sensor nodes into LIME. TinyLIME is implemented on top of LIME, internally using two tuple spaces, one for mote data access and one for issuing queries and command to the motes. On base stations, an interface is installed translating the tuple spaces to low-level mote communication. Sensor nodes run a TinyOS component processing these packets. They are only visible inside TinyLIME if they are in vicinity to a base station. Finally, **TeenyLIME** [82] completely pushes the tuples spaces into sensor networks without requiring a base station any more. Unlike LIME, tuple spaces are only shared with immediate neighbors and not with all transitive connected nodes. TeenyLIME guarantees freshness of data using epochs, extends tuple matching patterns to support range matching and introduces capability tuples to allow for on-demand insertion of tuples into the tuple space. In **SPREAD** [83], the publisher of a tuple defines the area around the publishing entity where the tuple is visible. A read operation will only find the tuple if the reading entity is inside the defined area. Therefore, in SPREAD tuples are always associated with physical objects.

Beside the tuple space abstraction, other systems provide data sharing in neighborhoods or groups, as well. **Hood** [490] is such a system for TinyOS to share data with one-hop neighbors. Local data is broadcast by Hood to the neighbors, which decide using a filtering interface if the sender node and the attributes should be cached locally. And node can also have more than one logical neighborhoods, e.g., one for routing information and one for sensor data. Therefore, shared data can be asymmetric in Hood. **Abstract Regions** [485] defines operators to create a neighborhood based on geographic (e.g., nodes within a certain distance) or radio properties (e.g., N-hop neighborhood, k-nearest neighbors), but also to create topologies like approximate planar meshes or spanning trees. If data is published on a node it is distributed to all nodes in the regions (with exceptions on spanning trees), thus leading to a tuple space behavior. It is also possible to requested directly from a single node and to aggregate values with the same key in the whole region. Abstract Regions also allows applications to explicitly trade of resource consumption and accuracy of global operations. **Neidas** [259] extends the cross-layer framework TinyXXL, which allows to share cross-layer data between several TinyOS components, with neighborhood data sharing. Accessing data from neighbors is transparent to the application; such data appears simply as an array with the ID of the neighboring node as array index. Neidas is pull-based and makes use of overhearing requests and data to save energy.

While the former approaches rely on physical properties to define neighborhoods, static and dynamic characteristics of nodes can also be used to define logical neighborhoods. In **EnviroTrack** [2] aims at object tracking applications. Using an application function, the middleware evaluates a sensor pattern and decides if a node belongs to a labeled logical group or not. A group leader is elected by the middleware that receives configured data of all nodes in the group. The middleware performs automatic aggregation and calls periodically a user-defined function that can use the aggregated state. **SPIDEY** [319] is a language to define logical neighborhoods using predicates over node characteristics. It also provides communication mechanisms to send a message to the neighborhood and to reply back to the sender of a message. SPIDEY allows to specify user-defined cost functions and a cost limit to tune between accuracy and resource consumption of the communication.

3.4.2.9 Service Invocation

In contrast to the mostly data-centric nature of sensor networks a service-centric paradigm is often used for Pervasive Computing applications. Therefore, communication between devices is not a simple data exchange but a remote procedure call. Due to the heterogeneity typically encountered in these scenarios interoperability and portability are major goals of middleware systems.

CORBA and **Java RMI** and the subsets for resource-poor devices **imumCORBA** and **J2ME RMI** Optional Package for the Connected Device Configuration are typical examples. The interfaces are specified using an interface definition language. An IDL-compiler creates so-called stubs and skeletons from it. When a stub is called on the client side, the middleware serializes the parameters, transmits them over the network, deserializes them at the server and calls the skeleton, which is implemented by the programmer and performs the actual task. Name services allow to find devices that offer a specific interface. **Universal Plug-and-Play** (UPnP) targets specifically pervasive computing environments like home networks. It builds on Internet protocols like TCP/IP, HTML and SOAP and encodes messages in XML. New devices advertise itself and describe the service interfaces they offer.

BASE [31] is a micro-broker middleware with only minimal functionality, i.e. accepting and dispatching requests. The actual protocols and communication technologies are implemented as plugins. BASE negotiates protocols and technologies with the communication partner based on capabilities and requirements selected by the application. Re-selection is done if a technology becomes unavailable. **dynamicTAO** [381] allows to transfer components in the distributed system, to load and unload components into the ORB during runtime, and to inspect and change the state of the ORB. Such components implement strategies, i.e. functional aspects like scheduling, concurrency, connection management or request demultiplexing. A minimal ORB is always running while the update of strategies takes place.

3.4.2.10 Event Detection

The Event Detection paradigm is particularly well suited to provide a programming abstraction for sensor networks applications since events are a natural way to represent state changes in the real world and in distributed systems, giving rise to model applications as producers, consumers, filters, and aggregators of events. There exist basic events, based on a simple real-world observation, and compound events, which are event patterns. If an event is detected applications that have shown their interest are notified. **DSWare** [272] is a software framework that provides several data service abstractions like data storage and caching, group management, data subscription, and scheduling and supports the specification and automated detection of compound events. A compound event specification contains, among others, a detection range specifying the geographical area of interest, a detection duration specifying the time frame of interest, a set of sensor nodes interested in this compound event, a set of basic events, a time window during which all basic events must occurs, a confidence function that maps all basic events to a scalar value and a minimum confidence value, which is the threshold for an event to be detected. DSWare uses and SQL-like interface for registering and canceling events.

TinyDB [143] also has event detection and signaling features embedded in its SQL dialect although its main functionality is the provision of a network database view. **Semantic Streams** [491] is a framework that supports queries over semantic interpretation of sensor data. It is based on event stream, which are flows of asynchronous events, and interference units that operate on these streams and generate new event streams with more semantic information. It is also possible to specify quality of service constraints, e.g., latency, power consumption or data quality, to select from several available input streams providing the same semantic information. Since Semantic Streams is implemented in Prolog the framework runs on PCs.

3.4.2.11 Context Management

Distributed Cooperating Objects systems are designed to measure properties of the physical world. They are, therefore, suitable for gathering the context of an entity, which is the information that can be used to characterize its situation. Individuals, locations, or any relevant objects can be such entities. Since a reasonable amount of data is collected in large systems, context management systems are needed to handle them. Such systems can separate applications from the process of sensor processing and context fusion. This context can be either queried by the application or it is used by other middleware functionality, mainly for adaptation purposes.

The **Context Information Service** [216] (CIS) of the Aura project models the relations between the entity classes people, devices, physical spaces and networks. CIS accepts SQL-like queries from a client that can also include QoS attributes. The queries are decomposed, underlying contextual information providers are contacted, and the result is synthesized. Information providers can be static, i.e. a database, or dynamic, i.e. an active component that tries to determine the answer upon request.

The **Nexus** [189] project aims at creating a large-scale augmented model of the real world that comprises real world objects like roads, buildings, room, people, cars etc. as well as virtual objects like virtual post-its attached to real world locations. A special language (Augmented World Modeling Language) is used to model all objects based on a hierarchical class schema. The augmented world can be queried using the XML-based Augmented World Query Language. The Nexus middleware decomposes the query and distributes the sub-queries to underlying model servers that should have the relevant information according to the Area Service Register. Nexus does not specify how the model servers are filled with data.

3.4.3 System Integration

In testbeds or experimental deployments, Cooperating Objects are regarded separately, i.e. the developer interfaces directly on a low abstraction layer with them. However in an operational deployment, the Cooperating Objects has to be included in a bigger context of – mostly existing – front-end software that can, for example, control or query it and receives in turn notifications or answers to the queries.

Shaman [410] is a Java-based service gateway that integrates resource-constrained sensor nodes into heterogeneous ad-hoc networks. The current system provides a Jini interface for the integration into Jini communities and a Java-applet based and HTML based interface for administrative purposes and direct human interaction. The interface is installed on the gateway host that acts as proxy for the sensor nodes. When a sensor node connects to the gateway it submits its service attributes that enable the proxy to provide a corresponding service. The gateway implements a request queue to support multiple client connections to the same service when the underlying sensor node supports only one connection.

Sensation [174] introduces a Sensor Abstraction Layer that hides the heterogeneity of different sensor platforms. It includes special drivers for each sensor network. Applications communicate with the abstraction layer using Unified Sensor Language (USL), which is based on XML. A Profile Registry stores the configuration and capabilities of the connected sensor networks. This information is used to discover which network can process which request. Also, an offline Data Storage can be used to answer requests on historical data or statistics. Two Java interfaces are provided based on the concept of queries and on the concept of location and devices of interest.

Sensor Andrew [388] is a large research project at Carnegie-Mellon which aims to build a campus-wide sensor networks. It has currently more than thousand sensors and nodes are organized in a three-tiered architecture. In each sensor network, there is a gateway; each sensor node in this sensor network conveys sensor readings to its gateway. A sensor reading is formatted in XML into a so called event node which is published at a server. This communication between the gateway and the server is performed through XMPP, a middleware for publish-subscribe communication and hence the gateway is a XMPP client and the server is an XMPP server. **AWARE** [19] is a EU-funded project which has created a middleware for cooperation between mobile and fixed sensors. The middleware offers publish-subscribe communication through channels; a *channel* is analogous to an event node in Sensor Andrew but it also offers the the extra feature that if no subscriber exist for a channel then a publisher do not publish data. **SensorMaps** (formerly called SenseWeb [396]) is a Microsoft supported initiative to create software to (i) export sensor readings over the Internet, (ii) allow visualization of them and (iii) help users to find a sensor networks in an area (by panning a map). SensorMaps does not aim to to perform processing of information from two or more sensor networks and hence it does not perform scalable data processing in the way that GSN does. SensorMaps allows however sensor readings from two different sensor networks to be overlayed in the presentation at that client web browser.

3.4.3.1 Database View

While the former approaches define how a Cooperating Object can be integrated into a bigger environment, the following approaches come from the other side and provide a generic and well-known abstraction for a Cooperating Object: a database view.

Both **Cougar** and **TinyDB** (see [143] for a description of both) provide an easy SQL-like interface to retrieve data from a wireless sensor network. Both systems optimize the query before disseminating it by evaluating different execution plans. Additionally, Tiny DB has extensions to repeat a query regulary or based on an event, to specify the lifetime of a query and to define events based on sensor data conditions. More systems, like SINA or DsWare, include SQL-like command as part of their interface. **PerSEND** [457] maintains a federated view of a relational database from the data available on proximate PDAs and provides an SQL-like interface to the applications. The database view is dynamic in the sense that it directly reflects a physical context. This context is represented by the set of near-by objects. As objects moves, the context evolves and the data associated to the objects are added or deleted from the database view. This system relies on a decentralized architecture, using only peer to peer communications (one-hop) over short distance wireless interfaces.

IRIS [148] (Internet-scale Resource-Intensive Sensor services) consists of a global collection of Organizing Agents, which are PC-class devices, and Sensing Agents, which are less powerful devices like PDAs. Organizing Agents upload scripts to the Sensing Agents to program gathering, preprocessing and sending data. The Organizing Agents build a global distributed database by indexing, aggregating, archiving and mining the data coming from Sensing Agents. To the user, data in IRIS is stored in an XML database that can be accessed using XPATH queries.

GSN [5] (Global Sensor Network) works with "virtual sensors", which is a data stream that is produced from an arbitrary number of input streams. Metadata for identification and discovery of the virtual sensor, the description of the input and output data streams and the temporal specifications of the virtual sensor, e.g., time window for input streams or history size and data rate of output storage, are described in XML. All data processing, i.e. the processing of the input streams and the combination of the input streams to form the output stream, is expressed using standard SQL. Input streams can be based on remote virtual sensors, which are obtained from other GSN nodes over the network, or local data sources, e.g., a connected sensor network or webcam. These are accessed with "wrappers", small program that act as interface between the data source and GSN. GSN is explicitly designed that data sourced can be owned by different organizations.

3.4.4 Debugging and Management Tools

3.4.4.1 Diagnosis

Diagnosis is concerned with analyzing a deployed system to detect faults and insufficient performance and to help the user identify the underlying causes. We consider three different classes of approaches. With active inspection, an existing system is modified to allow its diagnosis. With passive inspection, the goal is to not modify an existing system, for example by deploying additional nodes to overhear network traffic. A self-monitoring system includes mechanisms for diagnosis and repairing problems by design.

Active Inspection Current practice to inspect a deployed sensor network requires *active* instrumentation of sensor nodes with monitoring software and monitoring traffic is sent in-band with the sensor network traffic to the sink.

Nucleus [456] is a management system for sensor networks. It allows to query sensor node attributes over the network and provides a logging framework that delivers important local events to the sink. By querying for example the neighbor table or the state of the routing module, the networking behavior of nodes can be monitored and inspected in a live setting.

Sympathy [367] is a system for the detection and debugging of faults based on statistical data collected by individual nodes and forwarded to the sink node. It supports a fixed set of statistical metrics related to networking and makes use of the neighbor and routing tables as well as the number of packets received (correctly vs. with bit-errors) and transmitted. In case of a fault, e.g., if no data is received for a node in a certain period of time, the system uses a heuristic decision tree to infer the most likely root cause of the fault.

Memento [387] focuses on the efficient monitoring of the state of nodes and in-network failure detection for dead nodes and network partitions. The failures detection algorithm is designed to be robust to packet loss. Besides *node dead*, other binary states are reported, e.g., *low battery* and *network congested*. Compared to Sympathy, it is less flexible but it reduces the network traffic for monitoring significantly.

A Deployment-Support Network [111] is a second network which helps with the deployment of wireless sensor nodes. In this approach, each sensor node is connected physically to a deployment-support node which provides the functionality of a testbed but without a fixed network infrastructure. Instead, the reliable Bluetooth Scatternet of the BTnodes[46] provides a wireless back-channel and enables remote control of the sensor nodes. This effectively creates a wireless testbed and allows the sensor nodes to be deployed without additional restrictions. Although the deployment-support network approach allows to inspect a deployed sensor network, the fact that sensor nodes need to be physically wired to DSN nodes (requiring as many DSN nodes as there are sensor nodes) limits this approach significantly.

The main advantage of active inspection is that it can provide accurate access to the internal state of wireless sensor nodes in their real-world deployment environment. Unfortunately, however, the active inspection approach has several fundamental limitations. Firstly, problems in the sensor network (e.g., partitions, message loss) also affect the monitoring mechanism, thus reducing the desired benefit. Secondly, scarce sensor network resources (energy, CPU cycles, memory, network bandwidth) are used for inspection. In Sympathy, for example, up to 30% of the network bandwidth is used for monitoring traffic. Thirdly, the monitoring infrastructure is tightly interwoven with the application. Hence, adding/removing instrumentation may change the application behavior in subtle ways, causing probe effects. As reported in the previous chapter, changes to a deployed network should be avoided, if possible, to reduce the risk of failure of the network. Also, it is non-trivial to adopt the instrumentation mechanism to different applications or sensor network operating systems. Memento, for example, assumes a certain tree routing protocol being used by the application and reuses that protocol for delivering monitoring traffic.

Passive Inspection Packet sniffing is a common technique for passive observation of wireless networks [185] and has also been applied to sensor networks.

SNTS [232] uses distributed sniffer sensor nodes that record overheard traffic in local Flash storage. After an experiment, the nodes are collected and the packet traces are transfered to a central server. In contrast to WIT and JIGSAW, where the underlying 802.11 packet format is standardized, SNTS decodes the raw packet dumps based on a text file that describes the packet format. As an example for a possible processing of the packet traces, the authors employed machine-learning algorithms to identify bad sequences of events, which lead to an observed bug in the protocol/system, allowing them to fix the problem.

SNIF [377] allows to interfere the network state from message traces collected with a sniffer network. Interference is implemented with a data stream framework. The basic element of this framework is a data stream operator which accepts a data stream (e.g., a stream of overheard messages) as input, processes the stream (e.g., by removing elements from the stream or modifying their contents), and outputs another data stream. These operators can be chained together to form a directed acyclic graph. There are general-purpose operators that can be configured with parameters (e.g., a union operator that merges two data streams) and custom operators which are implemented by a user for a specific inference task. Ideally, one should be able to implement a given inference task just by configuring and combining existing general-purpose operators. However, in practice it is often necessary to implement some custom operators.

PDA [383] introduces passive distribute assertions to detect failure caused by incorrect interaction of multiple nodes and provide hints on possible causes to a user. PDA allow a programmer to formulate assertions over distributed node states using a simple declarative language, causing the sensor network to emit information that can be passively collected (e.g., using packet sniffing) and evaluated to verify that assertions hold. This passive approach allows to minimize the interference between the application and assertion verification. Further, the system provides mechanisms to deal with inaccurate traces that result from message loss and synchronization inaccuracies.

LiveNet [386] is a sensor network tool for network dynamics analysis. It uses passive sniffer nodes that forward overheard packets on the serial port to a connected laptop computer or stores them locally in the flash memory. Using an out-of-band mechanism, traces are collected on a central server and merged based on the WIT approach. The main analysis described in [386] is the reconstruction of the spanning tree routing paths using statistical methods.

Self-Inspection In contrast to active inspection, we define self-inspection of sensor nodes as mechanisms that observe the behavior of the sensor node and correct or report deviations from specified behavior.

Finne et al. report from a surveillance deployment, where they observed that during radio transmissions a subset of the sensor nodes triggered the PIR sensor which caused unwanted false alarms [128]. To ensure the automatic detection of the nodes with the specific hardware problem, they designed a self-monitoring architecture that probes the hardware to detect the problem. Nodes with the problem reconfigure themselves to turn off the PIR during radio transmission. The self-monitoring architecture of Finne et al. also integrates Contiki's software-based on-line energy estimator [107]. By comparing the measured power consumption with energy profiles described by the application developer, problems such as the CPU not going into low power mode can be detected and reported.

3.4.4.2 Healing

Healing is concerned with repairing problems in a deployed system once they have been detected. Healing can be performed at the protocol level, by updating the software executing on the nodes, or at the physical level, for example, by relocating nodes.

Self-Healing Communication Protocols In the area of mobile ad hoc networking, a number of protocols have been developed that deal with scenarios of moving and failing nodes. One of the most well-known protocols, the Ad hoc On-demand Distance Vector (AODV) protocol [351], is used in ZigBee. Other sensor networking protocols are able to circumvent links that are temporarily of bad quality. For example, MintRoute [495] monitors link conditions and continuously estimates the expected transmissions required to reliably reach each neighbor. Based on this information, MinteRoute selects the path with the least expected transmissions. Another technique used in sensor networking to avoid broken links and failed nodes is to use multiple paths [138].

Reprogramming When a deployed system is not working as expected, for example, due to a software bug, the system needs to reprogrammed. After deployment when no backchannel is available reprogramming needs to be performed over the air using a code distribution program. Distributing code updates is an energy-consuming task. Since the energy consumption increases with the size of the distributed code, modern sensor node operating systems such as Contiki [105] and SOS [166] have a modular structure which avoids the need to distribute the whole binary image.

The research community has developed a number of energy-efficient code distribution mechanisms including Deluge [200], Trickle [267] and MOAP [443]. Other researchers have proposed methods for reprogramming sensor nodes using image replacements [211], virtual machines [268], and version deltas [212, 239, 297, 373]. Both Contiki [105] and Mantis [459] provide dynamic linking based on the ELF format. Tsiftes et al. have shown the benefit of compressing dynamically linkable modules for reprogramming sensor networks [460]. Their experiments have demonstrated that compressing code modules reduces dissemination time and energy compression even though decompression on the sensor nodes requires processing time and energy.

Physical self-reconfiguration In cooperative systems it is of fundamental importance that the network is able to self-restore connectivity. In fact, critical information have to be provided to the devices responsible for taking the correct counter measures. However, embedded devices are often limited in terms of energy so that they may fail because of drain of battery. Furthermore, these devices can be deployed over a large, unattended, possibly hostile area. Thus, embedded devices are exposed to the risk of being damaged or even compromised. Such failures may cause network partitioning that the routing protocols could not be able to cope. Finally, embedded devices usually rely on wireless communication in order to simplify deployment and increase reconfigurability. Such wireless communication are unreliable due to presence of obstacles that can deteriorate or even nullify metrics of the Quality of Service.

The problem of network partitioning is not entirely new even though so far has received limited attention [428]. Many authors focus on cooperative networks composed of low-end cost-effective embedded devices responsible of monitoring the environment, such as Wireless Sensor Networks (WSNs). Chong and Kumar raise the problem of WSN partitions with a security focus [74]. So do Wood and Stankovic with respect to denial of service [497]. In [63], Cerpa and Estrin propose methods to self-configuring WSNs topologies. Although they mention the problem of network partitions as an important one, however, they leave such methods to future work. Finally, Shrivastava *et al.* propose a low overhead scheme to detect network partitioning, "cuts" in their parlance, but they do not propose any method to repair them [428]. In [100] authors propose a method based on autonomous mobile nodes. Once the network partitioning is detected, one or more mobile nodes equipped with a radio transmitter-receiver communicate with other devices. By reasoning upon the degree of connectivity with neighbors, a mobile node navigates to the partition gap to reach the optimal position to re-establish connectivity.

Boundary and Hole Detection An important aspect of repairing or healing is to detect the occurrence of holes and to identify their boundaries in the network structure. (Note that we refer to a boundary as either the outer boundary of the network or the boundary of connectivity holes.) This is particularly challenging if no information on the location of the individual nodes is available.

There exists a significant amount of work in the area of detecting the boundary of a sensor network without using location information [126, 133, 134, 255, 402, 482]. Common to all these approaches is that they work on the network connectivity graph and rely on certain given geometric properties. Based on the geometric information, these approaches can determine whether a node is an inner node or is on the network boundary. Additionally, some approaches can also provide information on the distance between a node and the boundary in the form of topological levels [255] or in the form of guaranteed minimum geometric distances [402].

To be able to differentiate between inner nodes and boundary nodes, the boundary/hole detection approaches make various assumptions on the node distribution or density of networks. Some approaches require a uniform distribution of nodes [126] while others assume that the length of the shortest path between two nodes provides a reasonable approximation of the geometric distance between the nodes [133, 134, 482]. The main limitation of approaches in this group is that they require a very high network density to provide reasonable result qualities.

Instead of making assumptions on node distribution or node density, some approaches assume a certain radio model, specifically the d-quasi unit disk graph model, for the communication among sensor nodes [255, 402]. Saukh et al. [402] describe a boundary recognition algorithm that collects connectivity information in the local neighborhood of a node and searches for geometric constructions, the so-called *patterns*. A node can detect that it is an inner node if it at the center of one of the possible pattern constructions. The simplicity of the pattern concept and the patterns themselves allows to find patterns for most inner nodes of a sensor network. This way, it is possible to narrow down the set of nodes that lie at the outer boundary of the network or at the boundary of holes.

3.5 Others

3.5.1 Modeling and Planning

Arguably, large-scale test-beds are the best way to design robust and efficient protocols for wireless Cooperating Objects (WCO) applications. However, the limited availability of test-beds have sparked the interest of the community in developing techniques and mechanism to *inform the design of a system prior to its deployment*.

These pre-deployment tools can be classified in 3 categories: a) analytical, b) simulation platforms and c) monitoring/management tools for small scale test-beds. These tools provide different levels of insight about the performance of a particular WCO application, but all of them share the same goal: identify potential risks that would severely reduce the quality of service perceived by the user (throughput, delay, lifetime).

In this section we survey generic pre-deployment tools; that is, tools that can be applicable to a variety of applications and/or scenarios (contrary to studies aimed to optimize the performance of a particular application in a particular scenario).

3.5.1.1 Radio Link Quality

Given that the performance of a network is fundamentally determined by its communication graph, it is central to consider a realistic representation of these graphs (node and link behavior) in the design of efficient algorithms. In this section we present a summary of the most important research work in the area link quality for Cooperating Objects.

Link Quality Characterization Wireless sensor network (WSN) protocols are often evaluated through simulations that make simplifying assumptions about the link layer. Arguably, the most popular model is the binary model, where nodes have perfect bidirectional communication within the circular radio range of the transmitter. However, experimental studies have demonstrated that the behavior of real links in low-power wireless networks (such as wireless sensor networks) deviates to a large extent from the ideal binary model. In real deployments links are unreliable, asymmetric, anisotropic, degree-heterogeneous and – depending on the environment – highly variable in time. A deep understanding of these effects is necessary due to the significant impact that they have on the performance of upper layer protocols.

Kotz *et al.* [242] enumerate the set of common assumptions used in MANET research, and provide data demonstrating that these assumptions are not usually correct. In one of the earliest works on a medium scale WSN test-bed (150+ nodes), Ganesan *et al.* [137] present empirical results on the behavior of a simple flooding in a dense Wireless Sensor Network . They found that the flooding tree exhibits a high clustering behavior (high degree-heterogeneity), in contrast to the more uniformly distributed tree obtained with the ideal binary model. Indoor and outdoor empirical studies by Zhao *et al.* [522] and Woo *et al.* [495] identified the presence of three distinct reception regions: connected, transitional, and disconnected. In the connected region, links are often of good quality, stable and symmetric. On the other hand, the transitional region is characterized by the presence of unreliable and asymmetric links; and the disconnected region presents no practical links for transmission. Unfortunately, the transitional region is often quite significant in size, and in dense deployments such as those envisioned for sensor networks, a large number of the links in the network (even higher than 50% [522]) can be unreliable. Similarly, the measurements obtained by the SCALE connectivity assessment tool [61] show that there is no clear correlation between packet delivery and distance in an area of more than 50% of the communication range. In [528], the authors provide a model to represent the anisotropic behavior of link in WSN. These works provided important initial insights on the particular characteristics of WSN links that are not captured by ideal models.

Impact on Protocol Performance Recent studies have shown that the unique characteristics of WSN links can have a major impact (both, positive and negative) on the performance of upper-layer protocols. In [242], it is argued that the real connectivity graph can be much different from the ideal disk graph, and the communication area covered by the radio are neither circular nor convex and are often noncontiguous. Similarly, Zhou *et al.* [528] reported that radio irregularity has a significant impact on routing protocols, but a relatively small impact on MAC protocols. They found that location-based routing protocols, such as geographic routing perform worse in the presence of radio irregularity than on-demand protocols, such as AODV and DSR. The negative effects were found to be particularly degrading in geographic forwarding schemes, as shown in [412].

Other works have proposed mechanisms to take advantage of nodes in the transitional region. The authors of [91] found that protocols using the traditional minimum hop-count metric perform poorly in terms of throughput, and that a new metric called ETX (expected number of transmissions), which uses nodes in the transitional region, has a better performance. Based on measurements for DSDV and DSR, over a 29 node 802.11b test-bed they show how ETX finds high throughput paths by incorporating the effects of link loss ratios, asymmetry, and interference. Along similar lines, Woo *et al.* [495] study the effect of link connectivity on distance-vector based routing in sensor networks. By evaluating link estimator, neighborhood table management, and reliable routing protocols techniques, they found that cost-based routing using a minimum expected transmission metric shows good performance. In [532], the authors analyze the positive effects of degree-heterogeneity on random walk-based queries.

Link Modeling Through empirical studies the previous works bring to light the impact that unreliable and asymmetric links have on protocol performance at different layers. Nevertheless, while an on-site deployment is arguably the best testing procedure for small-scale networks, it may be unfeasible for medium and large-scale networks, for which simulators are usually the best option. In order to help overcome this problem some tools and models have been recently proposed to obtain more accurate link-layer models.

The significant impact of real link characteristics on the performance of upper-layer protocols has created an increased understanding of the need for realistic link layer models for wireless sensor networks. In order to address this need, some recent works have proposed new link models based on empirical data. In [495], the authors derive a packet loss model based on aggregate statistical measures such as mean and standard deviation of packet reception rate. The model assumes a Gaussian distribution of the packet reception rate for a given transmitter-receiver distance. While this model was a first good approximation, later it was shown that the Gaussian assumption is not valid. Using the SCALE tool [61], Cerpa *et al.* [64] identify other factors for link modeling. They capture features of groups of links associated with a particular receiver, a particular transmitter, and links associated with a group of radios in close proximity. Using several statistical techniques they provide a spectrum of models of increasing complexity and increasing accuracy. A more recent model, called the Radio Irregularity Model (RIM), was proposed in [528]. Based on experimental data, RIM provides a radio model that takes into account both the non-isotropic properties of the propagation media and the heterogeneous properties of devices to build a richer link model. Motivated by this prior work, Zuniga *et al.* [533] proposed a probabilistic link layer model which captures unreliability and asymmetry. While the described work are important steps towards more realistic link layer models, most of them are focused on static environments, which do not capture the high variability of links in time, especially when considering mobile objects or mobile body sensor networks.

Temporal properties in dynamic environments have been studied in [62], the authors study short term temporal issues such as autocorrelation of individual and reverse links, and long term temporal properties such as the length of time the channel needs to be measured and how often to obtain accurate link quality metrics. The authors also propose new routing algorithms to take advantage of the temporal properties of wireless links.

3.5.1.2 Interference

Interference in wireless networks refers to the phenomenon where transmission between a pair or a set of nodes affects simultaneous transmissions between different pairs or sets of nodes. The bit error rate (equivalently, the ability of a node to decode a packet) depends on the signal strength of the received transmission as well as the signal strength of other simultaneous transmissions and the thermal noise at the receiver. The level of interference is quantified by Signal to Interference and Noise Ratio (SINR). SINR of a transmission from node s to some other node r is defined as

$$\mathrm{SINR}_{sr} = \frac{P_s}{\sum_{i=1,i\neq s}^{k} P_i + \text{Thermal Noise}_r}, \tag{3.1}$$

where P_i is the received power of the signal from transmitter i. The sum in the denominator extends to all interfering transmissions. The received power of the signal from some node i at node r is expressed by the path-loss formula

$$P_r = \mathsf{P}_i L d^{-\gamma}, \tag{3.2}$$

where P_i is the unattenuated transmission power, L is the average path-loss constant which is usually measured at 1 meter from the transmitter, d is the distance between i and r, and γ is called path-loss exponent. The value of the path-loss exponent depends on the physical environment, and commonly ranges from 2 to 6 [370]. Equation 3.1, however, ignores internal interference – the so-called multi-path channels. Multi-path refers to propagation of a signal through multiple paths of differing length, which results in multiple copies of the same signal separated by a time lag.

Equation 3.2 is an oversimplification and has been found inadequate [242]. In practice, modeling radio propagation in real-world situations is very complex. Consequently, measurement-based approaches have been proposed to address this problem. In a network of n nodes, the number of all possible node groups is 2^n. Clearly, one can not hope to measure all possible groups of links for interference unless the number of nodes in the network is very small. Padhye *et al.* propose a method that uses $O(n^2)$ broadcasts to estimate interference between unicasts [343]. Extending this work, Niculescu [334] present an expression for packet delivery ratio that decomposes the final delivery ratio between a pair of nodes as a product of delivery ratios when the interfering nodes act in isolation. Thus, one can estimate interference between any pair of nodes by taking $O(n^2)$ measurements.

The radio propagation is affected by variations in details of environment, such as temperature, humidity, presence/ disappearance of other electronic devices etc. Therefore, the use of measurements-only approach has limited applicability. Reis et al. advanced the state of art of this subject significantly by proposing a probabilistic *measurement based* *model* of radio propagation [374]. In their probabilistic model, Equation 3.1 becomes:

$$pr(\mathcal{A}_{sr}(P_{sr})) = \text{Prob}\left[\frac{\mathcal{A}_{sr}(P_{sr})}{I_r + \text{noise}_r} \geq \delta_r,\right] \tag{3.3}$$

where $pr(\mathcal{A}_{sr}(P_{sr}))$ is the probability that node r can successfully receive a packet from node s. The function $\mathcal{A}_{sr}()$ models signal strength attenuation. Accordingly, $\mathcal{A}_{sr}(P_{sr})$ is the attenuated signal strength of s at r. The interference experienced at r is I_r which is estimated from received signal strength (RSS) measurements.

As mentioned elsewhere in this document, the 2.4GHz radio frequency spectrum is shared by IEEE 802.15.4/ZigBee, 802.15.1 (Bluetooth) and 802.11b/g/n compliant devices. Although the use of different signal coding and modulation alleviates the extent of interference that may arise from devices using different standards, the proximity of a receiver to a high power transmitter can still cause interference.

3.5.1.3 Deployment Planning

Another interesting research direction related to Wireless Sensor Network (WSN) pre-deployment is WSN planning and development tools. Due to the special characteristics of WSNs such as unattended usage and limitations on energy consumption, computation capability, and communication bandwidth, the cost of deploying an efficient WSN for real application usage can be very high if any influential factor on the WSN performance has been overlooked. Thus, it is valuable to have a WSN planning platform to improve deployment efficiency, to reduce the deployment cost, and further to evaluate the WSN performance. Such a platform can help in solving problems of sensor node placement [208], node connectivity and coverage[24, 481], data collection, and the WSN evaluation.

While many network planning approaches have been proposed for Wireless Ad hoc Networks [500, 514], only little work on network planning has been done for WSNs. Li *et al.* [271] proposed a planning framework, POWER, which provides abstract WSN planning processes as well as actual framework models that facilitate network planning and evaluation. For such a platform tool to be effective, the key component is a simulation tool that provides various protocols/models to carry out the quantitative analysis of the WSN. In [25], the author identified a sequence of procedures to build a workflow of the WSN planning and deployment. The main steps include initial deployment, coverage validation, connectivity validation, communication protocol selections, and network evaluation by running the simulation. In addition, the author sketched a generic framework platform and implemented a basic simulation environment using J-Sim simulator.

From the very early stages of wireless sensor networks, scientists have been trying to increase the network functionality and reliability. To do so, people have created and proposed a big number of error correction and failure handling algorithms, since almost each study aimed to work on an already deployed network. Hence, most attention has been paid on getting the algorithms work better on the existing topology, rather than fitting the deployment to the needs of the application. Many of those algorithms increase the performance of the networks; however, at a cost of extra implementation, energy and time.

Nevertheless, there are some studies that target the physical deployment of the sensor networks. However, these algorithms basically aim to provide a better deployment for a better field sensor coverage. Also, they assume the nodes are able to automatically move at the run time of the application. Having the need of high-power processors and scalability are other major problems of those existing approaches. These drawbacks make them become inapplicable for the real-life scenarios.

3.5.1.4 Node and Network Lifetime

Prior to being able to discuss the lifetime of a wireless sensor network, one has to evaluate the individual node's battery lifetime. Here, monitoring the energy consumption or the battery state will be the basis, which lifetime prediction relies on. Battery effects which have to be taken into account for prediction, can also be exploited to actually maximize battery lifetimes. When extending the lifetime issue from node to network view, again, maximization approaches are an important aspect, but also power management and network lifetime prediction become increasingly important.

One kind of effective tools to estimate one individual node's battery lifetime is to target the energy consumption site: Landsiedel et al. [261] model each component of a sensor node in order to obtain a detailed power consumption model and match it with the applications running on the node. Alternative approaches try to safe this effort of modeling individual node components by accounting for the total current drawn from the battery, e.g. [364] or by monitoring the battery voltage, e.g. [486].

Batteries are non-linear systems: Their voltage does not linearly decline with the state of discharge. Furthermore, the rate capacity effect (disproportionate discharge with higher current), recovery effect (batteries recover charge when giving time to rest) and temperature dependency make the modeling of a battery a challenging task [215].

The accuracy of lifetime predictions based on data obtained by energy monitoring is strictly bounded by the correctness of the assumptions on the future usage of node. Thus, typically a constant usage is assumed, e.g. [486], [365],[368].

Watching an individual node is also important when targeting for lifetime maximization instead of lifetime prediction. Exploiting the battery effects stated above gives potential for battery lifetime maximization exceeding the potential of 'traditional' power (consumption) saving approaches. Approaches of battery aware task scheduling do exist in the context of embedded systems, e.g. [369],[260].

Extending the view from the individual nodes to the whole network, most research papers dealing with energy aspects are targeting a maximization of the network lifetime. A promising and widely used approach is to depend routing decisions on the node's battery states to unburden nodes which batteries are more depleted than others. While these approaches are only covering a few or a single aspects (here: routing), Jiang et al. propose

Tool		Models
Toilers-Code-Base	[55] [327, 328]	Random-Waypoint (sev. variants), Random-Walk, Prob. Random-Walk, Random-Direction, RPGM, Gauss-Markov, Column
BonnMotion	[463]	Random-Waypoint, Gauss-Markov, Manhattan-Grid, RPGM
Important	[23]	Random-Waypoint, RPGM, Freeway, Manhattan-Grid
MobiSim	[320]	Random-Waypoint, Random-Walk, RPGM, Gauss-Markov, Freeway, Manhattan
CanuMobiSim	[444] [435]	Brownian-Motion, Random-Waypoint, Meta-Model Obstacles
SUMO	[251]	Urban Vehicular Traffic

Table 3.1: *Tools for generating synthetic mobility traces*

an energy management architecture offering a generic interface [213]. They show, e.g., how such a system can be used to "manage resource usage by sharing system energy levels amongst nodes".

3.5.1.5 Modeling Mobility

Simulation and emulation are techniques frequently used for performance evaluation of Cooperating Objects. The movement patterns of the nodes are found to have significant impact on the simulation and emulation results. Various synthetic models were proposed during last decade. There have been several general surveys [22, 32, 55, 322] as well as some specific ones for vehicular models [192]. Instead of providing details concerning the different models, a table of tools to generate synthetic mobility traces is provided (cf. table 3.1). For all the tools listed it is possible to download a version on the respective website.

Bonnmotion Bonnmotion [463] is a Java software which creates and analyses mobility scenarios. It is developed at the University of Bonn, Germany, where it serves as a tool for the investigation of Cooperating Object scenario characteristics. The scenarios can also be exported for the network simulators ns-2, GlomoSim/QualNet, and COOJA.

Currently, there are five mobility models available: Static, Random-Waypoint, Gauss-Markov, Manhattan-Grid, RPGM. For scenarios analysis different metrics can be calculated as *overall* statistics (averaged over the simulation time) and as *progressive* statistics (values of metrics for certain points in time). The following metrics are supported: relative mobility, average node degree, number of partitions, degree of separation, average link duration, average time to link break.

3.5.2 Testbed and Simulation Platforms

Simulation and testbeds are indispensable tools to support the development and testing of Cooperating Objects. Simulations are commonly used for rapid prototyping which is otherwise very difficult due the restricted interaction possibilities with this type of embedded systems. Simulators are also used for the evaluation of new network protocols and algorithms. Simulations enable repeatability because they are independent of the physical world and its impact on the objects. Simulations also enable non-intrusive debugging at the desired level of detail. However, it has been shown that the models used for mobility, traffic, and radio propagation have a significant impact on the simulation results. Thus, appropriate models and model generators should be used.

Since simulations usually use over-simplified models of e.g. the radio environment, it is not enough to test cooperating object applications in the simulator only. Before deploying an application, testbeds are used as an intermediate step. Testbeds commonly consist of a large number of sensor nodes that are provided with permanent power supply and a back-channel for logging and control. The testbed infrastructure allows for nodes to be programmed and controlled (on/off, reboot) and provides a wired back-channel from each node, such that sensor nodes can be instrumented to send status information to an observer. As the behavior of a node and particularly its radio module is not simulated, testbeds provide a far more realistic behavior than simulations, but do not scale to large numbers of nodes.

The next sections present the state of the art for testbed and simulation platforms for Cooperating Objects. We divide simulators into general simulators, simulators specifically developed to simulate Cooperating Objects as well as emulators that simulate at the instruction level. Although some of the platforms described are also oriented towards robotics, they are widely used as simulators and testbeds for interoperability of heterogeneous Cooperating Objects.

3.5.2.1 Generic Simulators

Generic simulators that are used for simulating cooperating objects include OMNeT++, OPNET, NS-2 as well as GloMoSim and Qualnet.

OMNeT++ OMNeT++ [469] is a discrete event simulation package written in C++, primarily developed for the simulation of computer networks and other distributed systems. The OMNeT++ simulation models are composed of hierarchically nested modules that intercommunicate with message passing. Modules at the lowest level are programmed using C++, while the model structure is defined by a topology description language. Using this topology description language, modules can be combined and reused flexibly. OMNeT++ has also built-in support for the parallel execution of large simulations, although imposing some additional effort to the developer.

The package contains the C++ simulation kernel library, a manual, a simulation kernel API reference, a graphical topology editor, a graphical runtime environment with interesting animation and tracing capabilities, as well as a command-line runtime environment for batch execution. It also includes several other tools and sample simulations.

One of the strengths of OMNeT++ is that one can execute the simulation under a graphical user interface with interesting features [468]. The GUI makes the internals of a simulation model fully visible to the person running the simulation: it displays the network graphics, animates the message flow and lets the user to look into objects and variables within the model. The use of the tracing/debugging capabilities does not require extra code to be written by the simulation programmer.

OMNeT++ already contains detailed IP, TCP, FDDI and Ethernet protocol models, and several other simulation models. It forms the basis for the Mobility Framework/MiXiM described below that adds other interesting features. OMNeT++ is open source, free for non-profit usage, and has an active user community. It has been tested on Linux, Solaris, Windows and Mac OS X.

The OPNET simulation tool The OPNET Modeler is a discrete-event network modeling and simulation environment. It includes libraries for communication protocols such as the Transmission Control Protocol/Internet Protocol (TCP/IP), hypertext transfer protocol (HTTP), open shortest path first routing (OSPF), asynchronous transfer mode (ATM), frame relay, IP-QoS, 802.11, or Wi-Fi, and 802.16 or even WiMAX. These libraries provide the building blocks used to generate network (simulation) models. A network model consists of software objects that correspond to the devices, computers, and links that constitute the actual network of interest. The behavior of these objects is controlled by models of devices, computers, applications, communication protocols, and links. An OPNET Modeler project uses a three layer hierarchical architecture which encompasses the network model level (the highest level), the node and link model level and lastly, the process model level.

NS-2 Ns-2 is a discrete event network simulator [45]. It is popularly used in academic research for simulations of routing and multicast protocols over wired and wireless networks. ns-2 provides substantial simulation objects that support a wide range of network applications, protocols, and traffic models. ns-2 was initially developed as a variant of the REAL Network Simulator in 1989. ns-2 is currently supported by DARPA through the SAMAN project at USC/ISI and by NSF through the CONSER project.

ns-2 is an object-oriented simulator written in both C++ and OTcl. All network and protocol objects are organized into two hierarchies with one-to-one correspondence: the compiled C++ hierarchy and the interpreted OTcl hierarchy. The compiled C++ objects implement the actual definition and operation of protocols to achieve faster execution time in packet and event processing. The OTcl script, on the other hand, allows users to configure described network topologies and to specify interested protocols and applications for simulation experiments. Through an OTcl linkage, an OTcl script can invoke compiled C++ objects to achieve simulation efficiency. The execution of a simulation script in ns-2 is handled by a scheduler that manages a timing sequence of event objects. The scheduler uses an ordered data structure to maintain these event objects, and to execute each event at its specified time by invoking its corresponding event handler.

ns-2 provides flexible models at different network layers for constructing energy constrained wireless ad hoc network simulations. The mobile wireless networking environment in ns-2 also supports node movements and energy constraints.

SensorSim was designed as a simulation framework for modeling sensor networks built on NS-2 with some additional sensor network-specific features. It is however, no longer available.

NS-3 Ns-3 (www.nsnam.org) is a new software development effort focused on improving upon the core architecture, software integration, models, and educational components of ns-2. ns-3 allows the study of Internet protocols and large-scale systems in a controlled environment, and it is not backwards-compatible with ns-2. The ns-3 simulator borrows concepts and implementations from several open source simulators including ns-2, yans, and GTNetS.

GloMoSim and Qualnet GloMoSim [26] and its descendant Qualnet also include senor network models such as the physical and MAC layer of ZigBee.

Stage Stage [440] is an open source robot simulator, integrated in the open-source Player Project [355], capable of simulating a wide variety of platforms, robots and sensors on a 2D environment, see for instance [145] and [470]. Stage is designed to support research in multi-agent autonomous systems. It provides fairly simple, computationally-light models rather than attempting to emulate them with great fidelity. Stage allows rapid prototyping of controllers for real robots as well as simulating experiments with robots that are not physically available. Various sensors and actuators are provided in the Player Project distribution, including sonar, scanning laser rangefinders, vision (colour blob detection), odometry, and a differential steer robot base (odometry). Although it is mainly oriented towards robotics simulations, it also provides support to easily add new elements.

Player is a general purpose platform that facilitates the connection, communication and interaction among entities, [355]. Player is used as a link between the low-level hardware drivers in each entity with the high level software. The use of Player is widely spread in robotics community, see for instance [471] and [252].

Due to the compatibility between Stage and Player, few or no changes are required to move from simulation to practical test in hardware.

Gazebo Gazebo [142] is essentially an extension of Stage for simulation of 3D environments. Also open source under GNU Public License [236], the main difference between the Gazebo and the Stage is that Stage is designed to simulate large robot populations with low fidelity while Gazebo is designed to simulate small populations with high fidelity. Thus, both simulators (Stage and Gazebo) are complementary and are also compatible with Player. Therefore, Programs written using one simulator can usually be run on the other with few or no modifications and can also be tested in real hardware with very few modifications.

Gazebo generates both realistic sensor feedback and physically plausible interactions between objects since it includes a rigid-body physics simulation engine. It provides skins for reality augmentation of simple geometric models. Modules for simulation of a wide variety of robotic platforms and sensor, such as cameras, form the main manufactures are also included.

NetLogo NetLogo [452] is a multi-agent programming language and integrated modeling environment. The NetLogo environment comes with an extensive models library including models in a variety of domains such as economics, biology, physics, chemistry, psychology and many other natural and social sciences. It is particularly well suited for modeling complex systems developing over time. Modelers can give instructions to hundreds or thousands of independent "agents" all operating concurrently. This makes it possible to explore the connection between the micro-level behavior of individuals and the macro-level patterns that emerge from the interaction of many individuals.

3.5.2.2 Specialized Simulators

Lancaster Simulator The Lancaster hybrid test and simulation environment [316], shortened to LSE hereafter, is an architecture that supports the integration of third party simulators to evaluate location-based applications. Support is provided for real applications that can interact with simulated environments using a Web Services interface. The control and interaction of both simulators and applications are mediated and controlled by a Systems manager that is also responsible for experimental control.

LSE is built upon a distributed architecture which mediates the control and execution of applications and third party simulators. A System manager controls the execution of simulators and applications using a Web Services interface allowing these to be executed on separate machines. The authors integrate a popular network simulator, ns into their test environment. This is done without modification of the application code using a simple technique. Packets generated by the application are intercepted in a modified kernel and redirected through the ns simulator. Realistic application behavior is achieved by similarly intercepting application packets, which are then modified to receive location data from the simulator as opposed to from a real live location service. The test environment does not focus on simulation of large scale systems, nor on the simulation of hardware devices such as sensors and actuators instead leaving this task to a third party simulator.

Ubiwise Ubiwise [28] was one of the first simulators specifically developed for pervasive computing and was motivated by the need for rapid and cheap prototyping of pervasive devices and services. The simulator provides "a three-dimensional world, built on the

Quake III Arena graphics engine, and serves to simulate a first person view of the physical environment of a user". This simulated deployment enables the testing of pervasive services, implementation of protocols and integration of devices. Essentially, UbiWise is a human-in-the-loop simulator, which allows users to virtually interact with simulated ubiquitous devices in a 3-d space. The simulator aims to mix simulated and prototype devices and services where possible.

UbiWise supports the execution of application code in the form of Java .class files. As UbiWise is a human-in-the-loop simulator and runs in "real-time", wireless networks are not simulated but are actually introduced through interaction with a live network interface. As this is a human-in-the-loop simulator, this is possible as a 'simulation' runs at the same speed as in the real world. However, modeling features of standard wireless network behavior such as latency or interference is not provided as these factors exist naturally within a real network. These aspects are of course out of the control of the simulator itself.

NetTopo NetTopo [429] is an open source research-oriented simulator and visualizer designed to test and validate algorithms for wireless sensor networks. The goal of NetTopo is to build a sensor network simulation and visualization tool that gives users extraordinary flexibility to simulate their specific algorithms and is a compelling replacement of commercial simulator focusing on visualization of the communication in real Wireless Sensor Network test bed. Currently, NetTopo is released on SourceForge, and it has more than eighty Java classes and 11,000 lines of Java source code. Users can freely download the latest version of NetTopo by accessing the NetTopo website [17]. Due to its implementation in Java, NetTopo is platform-independent. Furthermore, it is flexible and extensible.

The Mobility Framework The Mobility Framework is intended to support wireless and mobile simulations within OMNeT++. The core framework implements the support for node mobility, dynamic connection management and a wireless channel model. Additionally the core framework provides basic modules that can be derived in order to implement own modules. With this concept a programmer can easily develop his own protocol implementations for the Mobility Framework without having to deal with the necessary interface and interoperability issues. The framework can be used for simulating fixed and wireless networks, distributed (ad-hoc) and centralized networks, sensor networks, multi-channel wireless networks and many other simulations that need mobility support and/or a wireless interface.

MiXiM The Mobility Framework has been merged with other simulators, including ChSim [467], positif [262], and MAC Simulator into MiXiM (**Mix**ed **Sim**ulators). Although different in goal, these simulators had a lot in common and a merger was a logical step. MiXiM [238] provides researchers with a modular framework, so the user can pick e.g. a simple basic module for the MAC, a specific mobility model with waypoints for a sink node and then focus on testing the performance of a new routing layer. Modules can be changed in the configuration without the need to recompile. Despite ongoing development, there is a wide choice of modules available already, especially for mobility, localization and MAC layers. To ease the process of making new modules, all a user has to do is to make his new class inherit from the appropriate base module and override a handful of methods. For statistical analysis, modules can publish parameters in the utility module of a node. Other modules can then subscribe to the published data and transform this into readable and usable statistics.

There is ongoing development in the area of placing objects/obstacles in the world and their effect on radio propagation as well as in the area of battery models. Furthermore, support for running the same code on both simulator and hardware is being developed for the MyriaNodes.

TOSSIM TOSSIM [270] is the simulator framework of TinyOS [186]. The main difference between TinyOS and other sensor network operating systems is the use of the specifically developed programming language nesC [141] that builds component abstractions on top of standard C. The availability of the tailored compiler for this operating system is also used to enable the simulation support. TOSSIM uses a combination of directly transforming accesses to variables in the compile phase and substituting simulated objects for hardware-near components (e.g., the radio stack). A simulation framework based on an event queue complements these simulator-specific changes. The output of the nesC-compiler is an Ansi-C file, which can either be compiled for the target platform or – in case of simulation – for the host platform. One major limitation resulting from this architecture is the support of only one application: no heterogeneous network – neither with respect to hardware nor to software – can be simulated.

Together with TinyOS, TOSSIM has undergone a major redesign with the release of the Version 2.0. The target platform for TOSSIM 2.0 is micaZ. The radio stack is replaced by a simulation model which provides extensive possibilities to configure the links between each pair of nodes in terms of signal strength, receiver reception and noise. The noise modeling is based on findings by Lee et al. [264], which improves the simulation fidelity especially for noisy environments. The MAC model supports a number of options, e.g., for preamble length, bandwidth, detailed timings, that can be controlled by the user and default to values from the TinyOS radio stack. TOSSIM includes strong support for scripting by integrating Python and providing access to important simulation objects.

The principle of same-source simulation and the tight integration with TinyOS make TOSSIM a suitable choice for the evaluation of protocols and algorithms implemented based on TinyOS. However, the exclusive use of this operating system and the missing support for heterogeneous networks, power profiling and mobility simulation, limit the generality and applicability of this simulator significantly.

COOJA COOJA [341] is a flexible Java-based simulator initially designed for simulating networks of sensors running the Contiki operating system [105]. COOJA simulates networks of sensor nodes. A simulated node in COOJA has three basic properties: its data memory, the node type, and its hardware peripherals. The node type may be shared between several nodes and determines properties common to all these nodes. For example, nodes of the same type run the same program code on the same simulated hardware peripherals. Nodes of the same type are initialized with the same data memory, except for the node id. During execution, however, the data memories of the nodes will eventually differ after reacting to external stimuli.

COOJA can execute Contiki programs in two different ways: Either by running the program code as compiled native code directly on the host CPU, or by running compiled program code in MSPSim. COOJA is also able to simulate nodes developed in Java at the application level. Java-based nodes enable much faster simulations but do not run deployable code. Hence, they are useful for the development of e.g. distributed algorithms. Emulating nodes allows control and retrieval of more fine-grained execution details compared to Java-based nodes or nodes running native code. Finally, native code simulations are more efficient than node emulations and still simulate deployable code. Combining the different levels in the same simulation can give both an efficient simulation as well as fine-grained execution details on selected nodes.

COOJA has been used for rapid prototyping of wireless sensor network mechanisms and applications. Furthermore, it has been used for protocol evaluation. It is more general than TOSSIM in that is not as tightly coupled to Contiki as TOSSIM is to TinyOS.

Castalia Castalia [41] is a WSN simulator that can be used by researchers and developers who want to do early-stage testing of their distributed algorithms and/or protocols in a simulator that tries to realistically capture the whole system. "Early-stage" refers to the initial efforts to validate an idea, usually done in simulation. Real testbed experimentation should follow. Part of the realism of Castalia comes from realistic wireless channel and radio model, with a realistic node behavior especially relating to access of the radio. Castalia can also be used to evaluate different platform characteristics for specific applications, since it is highly tunable, and can simulate a wide range of platforms.

Castalia's main features are an advanced channel/radio model based on empirically measured data, detailed state transition for the radio, allowing multiple transmission power levels and a highly flexible physical process model. Furthermore, Castalia simulates sensing device noise, bias, power consumption, node clock drift, and CPU power consumption as well as resource monitoring that goes beyond energy consumption such as memory and CPU time. In addition, there is a Medium Access Control protocol with a large number of parameters to tune. Castalia is based on OMNeT++. It is not designed to run deployable code since its intended use is a generic simulator not tied to a specific platform.

3.5.2.3 Emulators

MSPSim MSPSim [119] is a Java-based instruction level emulator of the MSP430 microprocessor series. MSPSim targets both realistic simulation with accurate timing for use as a research tool, and good debugging support for use as a development tool. In contrast with CPU-level emulators, it emulates complete sensor networking platforms such as the Tmote Sky and ESB/2. MSPSim provides both debugging capabilities such as break points, watches, logging, and single stepping as well as statistics about the operating modes of the emulated components, statistics such as how much time the CPU has consumed in the different low-power modes.

MSPSim combines cycle accurate interpretation of CPU instructions with a discrete-event based simulation of all other components, both internal and external. MSPSim uses an event-based execution kernel that enables accurate timing while keeping the host processor utilization as low as possible. Before interpreting instructions, MSPSim executes all pending events in both event queues. Each queue handles events that are scheduled with a different perspective of time, with the first being based on CPU clock cycles, whereas the other is based on a high resolution clock.

Most of the internal components of the MSP430, such as the USART and the analog-to-digital converter use the event queue for clock cycles, while external components such as radio transceivers use the event queue for the the high resolution clock. The emulator provides a programming interface for integration with simulation frameworks such as COOJA. In addition, the emulator can be extended with new mote types through a mote interface and I/O interfaces that correspond to the MSP430 I/O ports and serial communication ports.

MSPSim is integrated into COOJA as the emulation layer, see Section 3.5.2.2. Through the integration with COOJA, it is possible to emulate networks of MSPSim-emulated sensor nodes. These networks can be heterogeneous consisting of both TinyOS and Contiki nodes which enables e.g. interoperability testing [120].

ATEMU and Avrora As MSPSim, ATEMU and Avrora are instruction level simulators but for the AVR controller [14]. Both use a hybrid approach: the operations of individual nodes are emulated and the communication between them is simulated.

ATEMU [357] supports the Mica2 platform, but support for EEPROM and data flash is missing. Since last release 0.4 is date March 31st, 2004, ATEMU can be considered as abandoned.

Avrora [453] supports Mica2 and MicaZ platforms and achieves better scalability and is about 20 times faster than ATEMU by implementing a different synchronization strategy between nodes but it is nevertheless as accurate. Avrora is only 50% slower than TOSSIM.

Avrora offers a unit disc graph radio model but does not support mobile nodes. For debugging, testing and profiling, it provides probes, watches and events that can be used in small user-provided Java classes. Additionally, a wide range of monitors is shipped that automatically track, for example, power consumption, radio packets or function calls and print changes during the execution and/or generate a report after the program has completed execution.

3.5.2.4 Testbeds

The primary goal of any Cooperating Objects (CO) testbed is to support the design, implementation, testing and evaluation of applications and protocols without forcing the investigator to make artificial assumptions about system components or the system environment (as often needed in analytical and simulation work). The design of testbeds for Cooperating Objects, however, is facing a number of challenges and constraints, some of them being:

- Cooperating Objects are large-scale distributed systems where each node can observe the network state only locally. For network debugging and testing purposes, however, often a global view on the network is required.
- The protocols used in these systems tend to be application specific, and there is (yet) no unified set of protocols or well-tested equipment, nor are there any standardized and established mechanisms and tools for debugging and testing protocols and applications.
- Cooperating Objects are not only influenced by the built-in behavior of its software and hardware components, but also by external stimuli, since these systems are really built to observe and to modify the state of the physical world.
- Cooperating Objects must handle ever-changing topologies because of nodes dying on energy depletion or new nodes being added. They must also live with other limitations like scarce bandwidth, small amount of per-node memory, etc. The testbed infrastructure must be capable to track this dynamics.
- Many Cooperating Objects applications are not homogeneous and rely on nodes with different capabilities (in terms of processing power, energy supply and memory). This results in very specific distribution of functions like high-performance computation, data aggregation, data storage, interfacing with external networks, etc., that need to be replicated in the testbed environment.

A successful testbed architecture needs to accommodate these specifics, in a scalable and cost-efficient way. To facilitate the design of the System Under Test (SUT), for example, it should be possible to implement different network architectures in the testbed, from which the designer can select the best one for his application. To support the implementation/test/debugging phase, functionality like node (re-)programming as well as collection, processing and displaying of debug data are needed. For evaluation purposes, users should have means to precisely and reproducibly control network topologies and to inject node faults. Besides these development-oriented goals the testbed should be scalable enough to support large-scale deployments with more than just a few dozen nodes while keeping the cost and management overheads at an acceptable level. In the following we provide a short overview of several important differentiating features among the current testbed frameworks and instances.

System Architecture

- *Number of hardware levels*: The testing functions can be organized in a single, flat architecture, or in several different, hierarchically arranged node classes. The test-related hardware can be completely or partially separated from the SUT-related hardware. Since an application can have one, two or more tiers, the obtainable testing coverage crucially depends on the alignment between the testing system and SUT architectures.
- *Number of software levels*: The test-related software can function in a centrally organized manner (even in cases where the test hardware possesses multiple levels), or it can be distributed over different hierarchical levels. For example, leaf testbed nodes can generate a stream of observed information, which is filtered/preprocessed in higher levels of the testbed hierarchy.
- *Functional partitioning*: In some testbeds it is possible to partition the network between different applications (and related observers)? This partitioning can be truly orthogonal or partial where some resources have to be shared (e.g. radio resources like frequencies / codes).
- *Naming and addressing*: For maximal flexibility, the testbed architecture should decouple between the addressing of the node in the SUT space and its naming in the testbed architecture. Testbeds also differentiate in the naming schemes that are exported to the testbed user.
- *Active control*: Virtually all testbeds will offer functions to passively observe the state of the SUT. However, they can differ in their ability to modify the state of the application while it runs.
- *Synchronization capabilities*: Is the testbed capable of performing (possibly distributed) synchronized actions or not? Synchronization can for instance be useful to ensure that SUT nodes initialize at (almost) the same time, or to insert distributed failures in a tightly controlled fashion.
- *Self-configuration*: How much manual work is needed to add new components to the testing infrastructure and to track changes in the configuration of the testing system?

Function Decoupling

- *Communication channel*: Does the traffic generated by the testing functions (observed states, control traffic) has to share communication channels with the application traffic? If so, we call this *inband-signaling*, otherwise we speak about *out-of-band-signaling*. For example, a single-tier network where testbed infrastructure nodes are used entirely for testing functions, is regarded as using out-of-band signaling.

- *Processor and memory*: Another precious resource besides bandwidth are the computational and memory resources of SUT and testbed infrastructure nodes. When a node runs exclusively application functions or runs exclusively testing functions, we denote this as *processor/memory separation*. Passive snooping, for example, would be an example of processor/memory separation. If both application and testing functions run on the same processor/memory, we denote this as *processor/memory sharing*.

- *External configuration*: Is it possible to configure the SUT nodes and to supply them with information which otherwise has to be obtained by running protocols within the application?. An example would be a means to configure an application node with its geographic position as derived from the testbed topology, instead of forcing the node to execute a localization protocol. Using testbed supported time synchronization of the SUT nodes is another typical example.

External Interfaces

- *Execution model*: Testbeds differ in the models of test run execution that is supported. In general, two levels of access are provided: an online (interactive) mode, that allows to observe and modify the SUT state during runtime; or offline (batch/scheduled mode) where the collected data (logs, trace dumps, etc.), are made available to the user only after the test run has been completed.

- *Configuration and control API*: What types of interfaces are offered to users for configuration and control of the test runs. Possible choices can be, configuration languages, remote procedure calls, remote method invocation or variable-write operations.

- *Data access API*: The data access APIs define how the test run data is made available to the end users. For example, they can be stored in a relation database, written into plain files, custom XML files, offered as an SNMP MIB tree, etc.

- *Customization*: Can users plug own filters, own control functions or other code into the testing functions to modify their behavior? And if so, which components of the testing software can be customized? And what is the programming model?

Access Policy, Economical and Ethical Issues

- *Access policy*: What is the access policy for the testbed resources in terms of the access groups, the priority between the different groups, and the levied access fees.
- *Openness*: Is the hardware and software architecture of the testbed made available to external parties. What is the licensing model, and can external parties easily replicate the setup to validate the obtained results?
- *Privacy protection*: Does the testbed contain provisions to 'protect the private sphere of individuals who are monitored with or without consent?

3.5.2.5 Testbed Directory

APE The *Ad hoc Protocol Evaluation (APE)* testbed [282] from *Uppsala University* is a full-scale testbed (no artificial attenuation of the RF signals) for comparative study of different MANET protocols. Node mobility is achieved through choreographed movement of human volunteers carrying the laptops containing the SUT wireless cards. The experimental work is supported by a set of logging and visualization tools. The trace collection is performed in a distributed fashion, relying on timestamps and an off-line aggregation step to recover the global ordering of the events. The APE software framework is publicly available under open-source license.

CMU-DSR One of the original full-scale MANET testbeds, the *Carnegie Mellon University's* testbed for evaluation of the Dynamic Source Routing (DSR) protocol consisted of 5 mobile and 2 stationary nodes The wireless nodes were comprised by an IBM Thinkpad laptop equipped with 900 MHz WaveLAN radio card. Mobility effects were evaluated by driving the mobile units in rented cars in an outdoor area with rough dimensions of $700\text{m} \times 300\text{m}$. The localization of the units was performed via GPS. One of the fixed nodes was used as gateway, connecting the test network back to the Field Office via a 2.4 GHz point-to-point link.

Casino Lab The *Casino Lab* WSN testbed [60] at the *Colorado School of Mines* consists of 52 Tmote Sky nodes, hung from the ceiling of a large open industrial space with concrete walls, pipes, ducts and fluorescent lighting. The dimensions of the room is $24.4\text{m} \times 12.30\text{m}$, and the nodes are deployed in a 4×13 irregular grid. The nodes are connected via USB to 26 Tmote Connect Ethernet bridges providing a wired out-of-band channel for control. The *TOSSIM Live* extension [304] to the TOSSIM simulator, allowing execution of simulations in real-time and their interaction with real testbed nodes was originally developed and validated on the Casino Lab. The testbed is not publicly available.

DES-MESH The *DES-MESH* is a hybrid wireless mesh and sensor network testbed being developed at the *Freie Universität Berlin* geared towards long-term studies [161]. It currently consists of 35 hybrid nodes installed in an office setting spanning three floors. The testbed architecture is organized in three tiers: backbone mesh routers, mesh clients and sensor nodes. The mesh routers are equipped with 500 MHz AMD Geode LX800 CPUs with 256 MB of RAM, and have three IEEE 802.11 cards attached via USB hubs. The sensor nodes have 60 MHz ARM 7 cores and Chipcon CC1100 transceivers in the 868 MHz band. The testbed management is realized by a combination of SSH-supported remote command execution and SNMP services. The experiment configuration and control is facilitated by a domain specific language called DES-CRIPT based on XML.

DSN The *Deployment Support Network (DSN)* is a testbed framework developed at *ETH Zürich* [111], that leverages a secondary multi-hop Wireless Sensor Network optimized for connectivity and reliability as a testbed backbone. The DSN-nodes forming this backbone network are in turn connected to the SUT nodes via custom wired interfaces. Currently supported SUT node platforms include the BTnode, TinyNode, Tmote Sky and A80. The testbed backbone is used for SUT image file distribution, for transfer of logging and debug data, and for sending direct commands to the SUT nodes. The operation of the testbed is controlled by a DSN-server that exports the DSN-services via XML-RPC and web based interfaces towards the testbed user. The instance of the DSN framework at ETH Zürich uses the BTnode platform and its Bluetooth radio for the backbone network. The current configuration of the testbed consists of 30 backbone nodes and 30 Tmote Sky and TinyNode SUT nodes.

EWANT The *Emulated Wireless Ad Hoc Network Testbed (EWANT)* [394], developed at the *University of Colorado at Boulder*, is a reduced-scale MANET testbed with emulated RF environment using in-line attenuation and RF multiplexing. Mobility is simulated by discrete switching between different antennas connected to the outputs of the 1:4 RF multiplexers attached to the wireless cards.

iWWT The *Illinois Wireless Wind Tunnel (iWWT)* [466] is a reduced-scale testing environment for wireless networks implemented in an electromagnetic anechoic chamber at the *University of Illinois at Urbana-Champaign*. The main aim of the testbed is to create a realistic scaled version of the wireless environment maintaining full control over all relevant parameters that affect the performance of the wireless network like obstructions, interferers, etc. Mobility is supported by placing the wireless hosts (laptops, PDAs, sensor nodes) on top of remotely controlled cars. The scaled wireless environment is constructed by combining the effects of several building blocks: Power control module, Multipath module, Doppler module and Scattering Module. Despite these efforts for complete control of the RF environment, repeatability of small-scale experimental results remains elusive due to intrinsic randomness in the evaluated protocols and object positioning errors [525].

Kansei The *Kansei* testbed at *The Ohio State University* [11] has been initially developed as a testing facility for the middleware services for the final demonstrator of the *Extreme Scale Wireless sensor Networking (ExScal)* project [12]. The ExScal demonstration is one of the largest hybrid deployments of sensor and wireless mesh networks ever attempted, with more than 1000 sensor nodes and more than 200 wireless mesh nodes distributed in a 1.3km $\times$ 300m area. The Kansei testbed is a testbed environment that replicates this heterogeneous architecture at a reduced scale. Its stationary array originally consisted of 210 dual nodes (a combination of one Extreme Scale Stargate (XSS) wireless mesh node and one Extreme Scale Mote (XSM) node) placed on a 15 $\times$ 14 rectangular grid with about 1 m spacing. The XSS nodes have IEEE 802.11b wireless cards, and the XSM nodes, derivatives of the UC Berkeley's Mica family, operate in the 916 MHz band. Recently, 150 nodes have been upgraded with Tmote Sky boards. In addition to the stationary array, Kansei also has a portable array of 50 Trio sensor nodes and a mobile array of 5 robots from Acorname Inc.. The software architecture of the testbed is organized around the *Kansei Director* that provides interfaces towards the basic services of the testbed like experiment scheduling, deployment, platform monitoring and management as well as creation and management of testbed arrays and configurations. The Kansei testbed is designed for shared usage, and has open access policy for members of the research community.

MiNT The *Miniaturized Wireless Network Testbed (MiNT)* [90] at the *Stony Brook University* consists of eight mobile nodes roaming in an 3.66m $\times$ 1.83m area. The mobile nodes are built from COTS hardware: Routerboard 230 mini PCs with 1-3 Atheros IEEE 802.11a/b/g cards, placed on top of iRobot's Roomba as the mobility platform. One of the wireless cards on each node, operated in RF monitoring mode, is dedicated to collecting traces that are transferred to a central node where they can be visualized in real time. The output from the remaining wireless cards is connected to low-gain external antennas via fixed signal attenuators providing about 60 dBm attenuation, limiting the communication range to about 0.6 m. The custom control GUI enables convenient node configuration, editing and execution of traffic generation scripts, mobility scripts and fault injection scripts. The GUI performs merging of the traces collected by the different nodes and extraction of different network statistics. MiNT also supports hybrid execution of unmodified ns-2 simulations over the MAC and PHY layers of the real testbed nodes.

Mirage *Mirage* [77] is a testbed management system developed by the *Intel Research Berkeley (IRB)* that applies the concepts of microeconomic resource allocation to the problem of allocating nodes in a sensor network testbed. Users submit bids that are specifying their interest in terms of the nodes and the time they would like to be granted access to, combined with the price they would be willing to pay. The system periodically runs a sealed-bid auction to determine the successful bids based on the demand/availability, while aiming to maximize the aggregate utilization of the testbed. The Mirage framework is used to manage a 100 MicaZ and 50 Mica2Dot node testbed at IRB premises.

Mobile Emulab The *Mobile Emulab* [214] is a robotic wireless and sensor network testbed developed by the *University of Utah* leveraging their widely used *Emulab* platform for running network testbeds. The testbed is comprised by four Garcia robots from Acorname, each carrying a Stargate node that is interfaced with a Mica2 node. The mobile nodes are roaming in an area of about 8m × 3.5m. The tracking and the identification of the nodes is handled by a vision-based tracking system, using six ceiling-mounted video cameras aimed down towards the floor. A central control daemon is responsible for plotting the movement paths of the robots, so that they can safely reach the user-specified end positions, while maneuvering around any static and dynamic obstacles encountered during the motion. In addition, the testbed is equipped with 25 static Mica2 nodes arranged on the ceiling in a rough 2 m grid and on the walls near the floor. The testbed management software uses the standard Emulab services to provide a batch queued or interactive first-come, first-served usage through a web-based, GUI-driven or programmable XML-RPC user interface.

MoteLab *MoteLab* [488] is a very popular testbed solution from *Harvard University*. In its original design, the testbed was comprised from Mica2 nodes, each connected to Ethernet backbone via dedicated Crossbow interface boards, providing TCP forwarding for the serial ports. The current deployment is one of the largest publicly accessible testbeds with 190 Tmote Sky nodes deployed over 3 floors of Harvard's Engineering building. The SUT nodes are connected to the testbed backbone via Tmote Connect units. The testbed server provides a web interface that lets users monitor the status of the testbed and register jobs. The system uses a quota system to limit the time that each user has at its disposal for the outstanding test jobs. At one given time, only a single user has control over the complete resources of the testbed. The jobs are started by a job scheduler that configures the SUT according to the job description (installing and configuring the SUT images, etc.) and starts logging of the SUT output to a local database. The users can also get raw data access by connecting to the TCP forwarded serial ports of the SUT nodes during the assigned job slot.

Motescope The *Motescope* testbed [318] at the *University of California, Berkeley* is an update of the *sMote* testbed installed in the Soda Hall. The original 78 Mica2Dot nodes in sMote have been replaced with MicaZ nodes in Motescope. The testbed provides convenient web interface for configuration and control of the experiments. The testbed has open access policy for the members of the academic research community.

ORBIT The *Open Access Research Testbed from Next-Generation Wireless Networks (ORBIT)* [342] at *Rutgers University* is massive indoor grid of 400 wireless nodes each equipped with two IEEE 802.11a/b/g mini-PCI cards, deployed in a 20 by 20 grid with 1 m spacing. The testbed provides full remote control over the nodes during the assigned time slots, including installation of custom system images. The topology control is performed by a topology generator that leverages a wired remote control infrastructure to switch some of the nodes on or off. Similarly, mobility is simulated by transferring the state of a "mobile" node from one grid node to another. ORBIT provides extensive libraries for collecting, analyzing and accessing measurement data.

Omega The *Omega* testbed [339] is another testbed at the *University of California, Berkeley*. It consists of 28 TelosB nodes, connected via daisy-chained USB hubs to the central control server. This wired back-channel is used for powering, programming and debugging of the SUT nodes. The testbed has open access policy for the members of the academic research community.

PowerBench *PowerBench* [171] deployed at the *Delft University of Technology* is capable of parallel recording of the power consumption of all SUT nodes via low-cost custom interface boards with shunt resistors and A/D converter that is sampled by a modified Linksys NAS device. The setup achieves sampling rates of about 5 kHz and resolution of about 30μA (after calibration). The current deployment consists of 24 TNOde nodes installed on the ceilings of 4 rooms and 8 Linksys NSLU2 devices. The software architecture includes tools for controlling the allocation of nodes to test runs, sampling and converting the power consumption output from the A/D converters and well as TinyOS library for `print`-like debugging over the serial port.

Re-Mote The *Re-Mote* framework [89] developed by the *Datalogisk Institut (DIKU)* at the *University of Copenhagen* is comprehensive testbed management suite with four main software components: mote control infrastructure, database scripts, web services and GUI client. The database component uses the MySQL relation database and tracks the static and dynamic state of the testbed through the mote, mote host site and user session models. The core of the architecture is formed by the mote control infrastructure that is implemented in C++, separated in a low-level Mote Control Host (MCH) and Mote Control Server (MCS) parts. The web services component provides a loose interface between the testbed services and the clients. The Java-based client interacts with the Java web services component and the MCS allowing users to authenticate to the testbed and to program and monitor the SUT nodes. The Re-Mote framework has been tested in a deployment at DIKU with 36 Freescale DIG528-2 development boards as SUT nodes connected to 5 host PCs.

RoofNet The *RoofNet* testbed [35] at the *Massachusetts Institute of Technology* is an experimental multihop IEEE 802.11b Internet access network comprised of nodes positioned on the roofs of various buildings in Cambridge, Massachusetts, covering an area of about 4 square kilometers. In the original design, each node was built out of a mini PC and a 802.11b card connected to a roof-mounted omnidirectional antenna. The cards are operated in simplified IBSS ad-hoc mode which does not use beacons. The RoofNet routing protocol *Srcr*, implemented in Click [317], uses a combination of link-state and on-demand mechanisms and selects routes that minimize the estimated transmission time. The RoofNet deployment demonstrated that urban mesh-based Internet access networks can provide sustained DSL-level performance. The original software framework for RoofNet was open sourced and serves as basis for many RoofNet replicas like the Berlin Roof Net [201] and the NetEquality Portland deployment [329]. The RoofNet technology was commercialized and is being further developed by Meraki [303].

Sensei *Sensei* [375] is a nomadic testbed that can be used in a lab, for in-situ experiments and as prototype deployments. Physical sensors are normally attached to a Linux host consisting of a stationary computer, a laptop, a broadband router or a PDA but there is also a cellphone implementation that enables using a cellphone as sensor proxy or as a sensor itself. The testbed supports usage mobile sensor nodes, carried around on random or predefined paths by either humans or robots. Control and management of the testbed can be handled remotely via a GUI that is highly customizable for different purposes. The nomadism supports testing the same experiment in different locations or at different testbed installations. The support for repeatable mobility enables repeating similar tests including mobile nodes multiple times, and the highly flexible GUI makes it possible to create experiment-specific visualizations of testbed activities.

Player *Player* is a GNU Public License open-source platform robot device interface and server. It is used as a network server for robot control and provides a clean and simple interface to the sensors and actuators of a robot over an IP network. Any User Program can communicate with Player over a TCP/IP socket, providing the User Program with functions to read data from sensors, write commands to actuators and configure devices [355]. Player is composed of two basic modules. Player Servers include the Player Drivers to interact with the hardware elements, for instance to read the sensors. Player Clients connect to the Player Servers and provide a clear interface (called Player Interface) with the User Program. All communications between processes are carried out transparently using TCP/IP. Thus, although the allocation of the Player Servers is local to the hardware devices, the Player Clients can be executed in any machine with TCP/IP access to Player Servers. Player supports a wide variety of hardware and includes drivers of a wide range of supported devices. Among the supported elements there are platforms such as MobileRobots PSOS/P2OS/AROS, sensors and other hardware elements. Besides, Player's modular architecture also allows adding support for new hardware, and an active user/developer community contributes with new drivers. Player is language and platform independent and makes no assumptions about how the robot control programs might be structured. The User Program can be a highly concurrent multi-threaded program or a simple "read-think-act" loop. Also, Player is compatible with Stage and Gazebo simulators. Few or no changes are required to move from simulation to practical test in hardware.

YARP *YARP* [508] is an open-source framework for distributed computation and inter-process communication. It provides a set of libraries, protocols, and tools that keep modules and devices cleanly decoupled and help to organize communication between sensors, processors, and actuators minimizing incompatibility problems between architectures, frameworks, and middleware. Its model of communication is transport-neutral, so that data flow is decoupled from the details of the underlying networks and protocols in use (allowing several to be used simultaneously). YARP is written almost entirely in C++ and is OS neutral and has been used on Linux, Microsoft Windows, Mac OS X and Solaris. The components of YARP can be separated into:

libYARP_OS which interfaces with the operating system(s) to support easy streaming of data across different threads across different machines. YARP uses the open-source ACE (Adaptive Communication Environment) library, which is portable across a very broad range of environments.

libYARP_sig which allows performing common signal processing tasks in an open may easily interfaced with other commonly used libraries, such as OpenCV.

libYARP_dev which interfaces with common devices such as framegrabbers, digital cameras, motor control boards, etc. The lib-YARP dev library is structured to interface easily with manufacturer-supplied code, but to shield the rest of your system from that code to facilitate future hardware replacements.

TWIST The *TKN Wireless Indoor Sensor Network Testbed (TWIST)* [169] is a multi-platform, hierarchical testbed architecture developed at the *Technische Universität Berlin*. The TWIST instance at the TKN office building is one of the largest remotely accessible testbeds with 204 SUT sockets, populated with 102 eyesIFX and 102 Tmote Sky nodes. The SUT nodes are deployed in a 3D grid spanning 3 floors of the office building. The TWIST architecture introduces a layer of "super-nodes" between the SUT and the testbed server that play role in modeling hierarchical and hybrid SUT setups and in decentralizing the testbed management functions. TWIST relies on COTS hardware and fully leverages the features of the USB 2.0 standard. The SUT nodes are connected to the super-nodes via USB hubs which act as concentrators and provide a power supply management capability. This enables active SUT topology control and node fault injection modeling through selective powering on and off of SUT nodes. The software architecture is designed for easy remote access. The primary user interface is web-based and provides support for registration, configuration and monitoring of test jobs. The testbed resources can be shared among multiple users as long as each accesses different SUT platform. The web-interface is coupled to a background testbed server that uses a RPC architecture to distribute the testbed management tasks to the super-nodes, where they are executed in parallel on the attached SUT nodes. TWIST provides automatic trace collection and centralized time stamping service, as well as raw access to the serial I/O of the SUT nodes.

Tutornet *Tutornet* [462] in the Ronald Tutor Hall at the *University of Southern California* is a tiered testbed with 13 node clusters, formed by a Crossbow stargate cluster-head, connected to several mote-class nodes via USB cables. The stargate tier uses the EmStar development platform [150] and operates as a multihop wireless network, using AODV-based routing over IEEE 802.11b links. The mote tier is currently populated with 81 Tmote Sky and 13 MicaZ nodes. The testbed has open access policy for the members of the academic research community.

WASAL The WASAL testbed [92] at the *École Polytechnique Fédérale de Lausanne* consists of 25 TinyNode nodes connected to a wired testbed back-channel via custom serial-to-ethernet devices that act as passive communication bridges and range extenders. The WASAdmin management tool uses an XML-based configuration language and concurrently executes a separate shell instance and script parser for each target node in the SUT.

WINTeR The *Wireless Industrial Sensor Network Testbed for Radio-Harsh Environments (WINTeR* [432] is a testbed facility for the Canadian *Petroleum Applications of Wireless Systems (PAWS)* project. It aims to replicate the harsh RF conditions of an offshore oil platform, where the dense piping and other large metallic structures create complex multipath effects, combined with strong noise and interference and challenging environmental conditions. To this end, a platform mockup can be crated out of six modules each with dimensions of 2.44m×1.83m×2.74m. The mockup includes typical industrial metal structures like beams, pipes and tanks. The testbed uses a customized node platform consisting of a mote, DC power source, embedded CPU, micro power meter and programmable attenuator placed in an industrial enclosure. The emulation of the RF environment is further supported by an VSG-based EMI generator. The software architecture of the testbed is based on a modified version of Harvard's MoteLab suite.

3.5.3 Standards

In 2005, ON World Inc. carried out a survey with 58 OEMs and platform providers in residential, commercial buildings and industrial markets and published it in the report "Wireless Sensor Networks – Growing Markets, Accelerating Demands". Among the results provided, one of the most interesting ones for the purposes of this roadmap are the reasons for late adoption of wireless sensor technology.

In that report, standardization was found to be the second most important barrier. The main reason for this is that most companies insist on interoperability and at that time the only standard was ZigBee. ZigBee is still among the dominating standards today but in the meantime a number of other standardization efforts have been made. The following presents the standardization activities and standards development organization (SDO) relevant for Cooperating Objects.

3.5.3.1 ZigBee and the ZigBee Alliance

The ZigBee Alliance is an association of companies working together to enable reliable, cost-effective, low-power, wirelessly networked, monitoring and control products based on an open global standard.

The goal of the ZigBee Alliance is to provide the consumer with flexibility, mobility, and ease of use by building wireless intelligence and capabilities into everyday devices. ZigBee technology can be embedded in a wide range of products and applications across consumer, commercial, industrial and government markets worldwide. Companies participating in ZigBee alliance work together to have a standards-based wireless platform optimized for the needs of remote monitoring and control applications, including simplicity, reliability, low-cost and low-power. Main objectives of the ZigBee Alliance standardization activities are the following:

- Defining the network, security and application software layers.
- Providing interoperability and conformance testing specifications.
- Promoting the ZigBee brand globally to build market awareness.
- Managing the evolution of the technology.

The ZigBee specification has been released and is publicly available on the ZigBee website (http://www.zigbee.org/en/spec_download/zigbee_downloads.asp). The ZigBee specifications include the description of the architecture and functionalities of ZigBee stack but also the Public application profiles that finalized the test program can be accessed and downloaded. The new ZigBee features include the following ones:

- Frequency agility
- Enhanced security
- Optimized routing for large networks
- Multicast

The finalized public profiles include Home Automation and Smart Energy applications. The ZigBee Alliance is working on defining the messages to be used by many other application scenarios. The following describes the status of groups and the activity that is currently underway:

- Home Automation (HA): the specification and Test Program have been finalized and are publicly available; the group is currently working for enhancement of the profile and for enhance the interoperability with other application profiles; this profile specifies the messages for controlling devices and sensors in the home environment.
- Smart Energy (SE): the group finalizes the specification and Test Program and several products have been certified as SE compliant so far; the group is now working on defining new features to be included in next releases of the profile and is also defining best practices for configuration and installation of the devices; this profile specifies scenarios like energy metering, demand response applications and load control.
- Commercial Building Automation (CBA): the specification has been finalized and the group is working on the test documents in order to be able to lead products towards the certification process; the profile defines the control and management of BACNET systems (there is a liaison between ZigBee and BACNET for this group) and the messages to operate a wireless control management system integrated in cable building management system.
- Telecom Applications (TA): the group finalized the specification and is working on the test program; the TA profile specifies applications that involve mobile handsets and ZigBee nodes such as the following ones.

  Mobile advertising and information delivery

  Multi player chatting and mobile gaming

  Indoor location-based systems

  M-commerce and mobile office

- Personal Home and Hospital Health Care (PHHC): the group is finalizing the specification and starts working on the test documents. The profile will relay on existing standards such as IEEE 11073 and is working in the Continua Alliance to promote the ZigBee technology.
- Wireless Sensor Application (WSA): the group developed the specification but is waiting for finalizing the discussion about low power router features before start working on the test documents.

Other relevant activities running within the ZigBee Alliance concern the finalization of the specification of Gateway. Moreover, the ZigBee Alliance created an IP connectivity

Application profile Name	Marketing documents	Technical Requirements	Specification	Testing Documents	Certified devices	Spec availability	Logos
Smart energy	Completed	Completed	Completed	Completed	YES	YES	
Home Automation	Completed	Completed	Completed	Completed	YES	YES	
Telecom Applications	Completed	Completed	Testing Phase	In progress	-	-	
Commercial buildings	Completed	Completed	Testing Phase	In progress	-	-	
Health	Completed	Completed	Testing Phase	In progress	-	-	

Figure 3.11: *The ZigBee Profiles for several application domains*

group in order to harmonize and collect the activities related to the integration of IP network.

ZigBee Future Developments The ZigBee Alliance is defining a roadmap for 2009 scope and work; there will be resources dedicated to the finalization and roll-out of emerging profiles with the definition of best practices to enhance easy-to-use and plug and play capabilities. Other relevant activities will be leaded for battery-less device marketing discussion and technical evaluation of incorporating this feature in the future ultra-low power devices. Moreover low power sleeping router techniques will be investigated and possible incorporation in the standard will be discussed. The Alliance will also work in order to create liaison with other standard bodies in order to avoid the fragmentation in multiple solutions; as for collaboration, a liaison with Home Plug initiatives will be started by end of 2009.

3.5.3.2 IP for Cooperating Objects and Smart Devices

While a number of years ago IP was regarded too heavyweight for resource-constrained devices there are now several activities towards IP for resource-constrained devices such as sensor nodes. The IPSO (IP for Smart Objects) Alliance promotes IP for such devices and in the IETF the 6LoWPAN and ROLL working groups define standards. Cisco, Atmel, and SICS recently announced uIPv6, the world's smallest open source compliant IPv6 stack, for the Contiki operating system [109]. uIPv6 passes all the tests required for an IPv6 stack to be called IPv6 Ready.

6LoWPAN – IPv6 over Low-Power Wireless Personal Area Network The IETF 6lowpan Working Group aims to define the transport of IPv6 over IEEE 802.15.4 low-power wireless personal area networks.

The most exhaustive description of the Working Group and of the state of the art of standardization process may be found on the IETF site. The following rundown comes from the charter page of the working group.

Description of the Working Group The Working Group has completed two RFCs: "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals" (RFC4919) that documents and discusses the problem space and "Transmission of IPv6 Packets over IEEE 802.15.4 Networks" (RFC4944) which defines the format for the adaptation between IPv6 and 802.15.4.

The Working Group will generate the necessary documents to ensure interoperable implementations of 6LoWPAN networks and will define the necessary security and management protocols and constructs for building 6LoWPAN networks, paying particular attention to protocols already available. 6lowpan will work closely with the Routing Over Low power and Lossy networks (roll) working group which is developing IPv6 routing solutions for low power and lossy networks (LLNs).

Routing Over Low power and Lossy networks (ROLL) The IETF ROLL Working Group aims to define a routing architectural framework for IP-based (IPv6) low power and lossy networks, regardless the adopted underlying transmission technology.

Once again the most exhaustive information source is the IETF site, so the following description has been drawn on the Working Group charter page.

Description of Working Group Low power and Lossy networks (LLNs) are typically composed of many embedded devices with limited power, memory, and processing resources interconnected by a variety of links, such as IEEE 802.15.4, Bluetooth, Low Power WiFi. LLNs are transitioning to an end-to-end IP-based solution to avoid the problem of non-interoperable networks interconnected by protocol translation gateways and proxies. In addition, LLNs have specific routing requirements that may not be met by existing routing protocols, such as OSPF, IS-IS, AODV and OLSR. For example path selection must be designed to take into consideration the specific power capabilities, attributes and functional characteristics of the links and nodes in the network.

There is a wide scope of application areas for LLNs, including industrial monitoring, building automation (HVAC, lighting, access control, fire), connected home, healthcare, environmental monitoring, urban sensor networks sensor networks, assets tracking, refrigeration. The Working Group will only focus on routing solutions for a subset of these. It will focus on industrial, connected home/building and urban sensor networks and it will determine the routing requirements for these scenarios.

The Working Group will provide an IPv6 only routing architectural framework for these application scenarios. Given the transition of this technology to IPv6, at this time it is believed that an IPv4 solution is not necessary. The Framework will take into consideration various aspects including high reliability in the presence of time varying loss characteristics and connectivity while permitting low-power operation with very modest memory and CPU pressure in networks potentially comprising a very large number (several thousands) of nodes.

The Working Group will explore aspects of mobility within a single LLN (if any) in the routing requirement creation. The Working Group will pay particular attention to routing security and manageability (e.g., self configuration) issues. It will also need to consider the transport characteristic the routing protocol messages will experience. Mechanisms that protect an LLN from congestion collapse or that establish some degree of fairness between concurrent communication sessions are out of scope of the Working Group. It is expected that applications utilizing LLNs define appropriate mechanisms.

IP for Smart Objects (IPSO) IPSO is a new alliance (formed on 16th of September 2008) the aim of which will be to promote the use of IP(v6) in sensors/objects networks by producing use cases and white papers and organizing some interoperability events. IPSO will exclusively rely on SDOs (IETF, IEEE, . . .) to specify technologies.

Up to now the newly formed Alliance has been mostly focused on the definition of its own mission and internal organization. The most relevant information coming from the official site are report in the following paragraphs.

Mission The Alliance is a global non-profit organization serving the various communities seeking to establish the Internet Protocol as the network for the connection of Smart Objects by providing coordinated marketing efforts available to the general public. Its purpose is to provide a foundation for industry growth through building stronger relationships, fostering awareness, providing education, promoting the industry, generating research, and creating a better understanding of IP and its role in connecting Smart Objects / Cooperating Objects.

Goals IPSO has the following goals:

- Promote IP as the premier solution for access and communication for Smart Objects.
- Promote the use of IP in Smart Objects by developing and publishing white papers and case studies and providing updates on standards progress from associations like IETF among others and through other supporting marketing activities.
- Understand the industries and markets where Smart Objects can have an effective role in growth when connected using the Internet Protocol.
- Organize interoperability tests that will allow members and interested parties to show that products and services using IP for Smart Objects can work together and meet industry standards for communication.

- Support IETF and other standards development organizations in the development of standards for IP for Smart Objects.

It should be noted that the objective of the Alliance is not to define technologies, but to document the use of IP-based technologies defined at the standard organizations such as IETF with focus on support by the Alliance of various use cases.

In establishing the Alliance one of the subsidiary goals will be to be as efficient as possible in environmental matters and our preference will always be to make appropriate substitutions such as virtual meetings for in-person meetings.

3.5.3.3 ETSI

Within ETSI there are two major initiatives related to Wireless Sensor Network issues: the first is the proposal for the creation of a Technical Committee on Machine to Machine (M2M) and the second one is the launching of a new Industrial Study Group on the topic of “Integrating the Physical with the Digital World”.

Both initiatives are in the setting phase so none of them is already something like a standardization group with defined roadmap and strategy.

The activity related to M2M came from the follow-up of an ETSI workshop on the topic held in June 2008; since the initiative had a good success it was decided to create an ad-hoc group with the objective of defining a proposal to submit to the ETSI

Board for the creation of a Technical Committee on M2M. This ad-hoc group (chaired by France Telecom) met a couple of times so far and it is trying to summarize and establish a standardization direction for the many topics of interest related to M2M; some of them are:

- APIs for application developers
- Gateway APIs Wireless and wireline access interfaces
- Network optimization
- Service layers
- Interoperability
- Testing
- Regulatory aspects
- Identification and addressing (MSISDN)
- Security and privacy
- Management

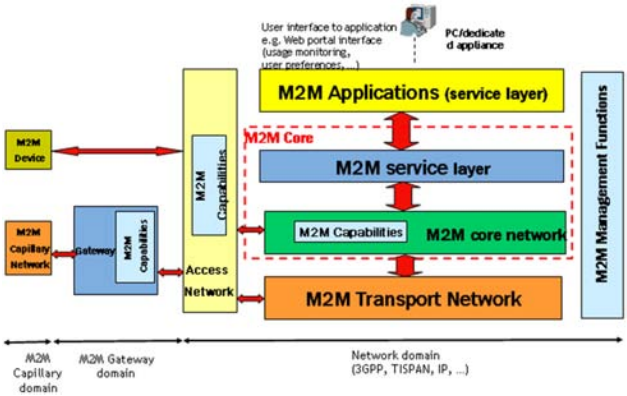

Figure 3.12: *M2M Architecture*

- Standardization and vertical applications use cases

An architecture of the M2M world is under discussion in order to identify the missing layers/interfaces/building-blocks that are standardized in any of the different standardization bodies. The results of the ad-hoc group that will consist in a proposal for the establishment of a new Technical Committee will be evaluated by the ETSI board by October 2008.

The other initiative is the proposal conducted by Telefonica for the establishment of an Industry Study Group on the physical world interface; within this group the idea is to start from the work going on in FP7 project SENSEI (www.ict-sensei.org) for the creation of an global framework and architecture for the creation, management and development of applications based on wireless sensors and actuators networks. The ISG would perform pre-standardization and specification activities on the following topics:

- Short Range Radio Mesh Networks devices and gateways of various kinds
- SRRMN connectivity, networking, security, AAA, plug/play capabilities, operation and management
- Functional Entities, architectures and protocols enabling the creation of valued added services founded on SRRMN Semantics and representation of sensor and actuators information models

- Application development interfaces
- Physical world interface architectural framework
- Management services for SRRMN islands

The ISG is in the setting phase, identifying specific target activities and partners for the ISG sponsorship, so for the moment no roadmap is available yet.

Another Technical Committee is working on an application domain specific for the automotive sector (ITS - Intelligent Transport Systems) which has several issues related to Wireless Sensor Networks but specific for the application domain.

3.5.3.4 Continua Alliance

The Continua Alliance (CA) is non-profit open industry alliance of the finest healthcare and technology companies in the world joining together to improve the quality of personal healthcare.

Started in 2006, the CA is chaired by Intel and includes more than 160 companies worldwide that are technology, medical device and health care industry leaders in the area of tele-health. Founding companies of the CA include BodyMedia, Cisco Systems, GE Healthcare, IBM, Intel, Kaiser Permanente, Medtronic, Motorola, Nonin, Omron Healthcare, Panasonic, Partners HealthCare, Polar Electro, Royal Philips Electronics, RMD Networks, Samsung Electronics, Sharp, The Tunstall Group, Welch Allyn and Zensys.

The mission of the CA is "to establish a system of interoperable personal tele-health solutions that fosters independence and empowers people and organizations to better manage health and wellness". The objectives of the CA include:

- Developing design guidelines for interoperability based on selected existing standards
- Establishing a product certification program with a consumer-recognizable logo signifying the promise of interoperability with other certified products.
- Collaborating with government regulatory agencies to provide methods for safe and effective management of diverse vendor solutions
- Working with leaders in the health care industries to develop new ways to address the costs of providing personal tele-health systems.

The CA does not aim at defining new standards. Instead, it selects existing standards and defines guidelines on how to use them to achieve interoperable tele-health solutions.

The application areas that are considered by the CA are: Health and Wellness, Disease Management, and Aging Independently.

3.5.3.5 WirelessHART

WirelessHART is an extension to the Highway Addressable Remote Transducer (HART) protocol. HART is a protocol for bi-directional communication between a host application and intelligent field instruments. HART is used diagnostics, remote process variable interrogation and parameter setting.

WirelessHART is an open communication standard especially designed to address the requirements of the process industry. The WirelessHART Communication Specification (HART 7.1) was approved by the International Electrotechnical Commission (IEC) as a Publicly Available Specification in September 2008. Simplicity, reliability and secure wireless communication are parts of the requirements that it fulfills. The protocol stack of WirelessHART contains a physical layer, a data link layer, a network layer, a transport layer and an application layer.

Physical Layer WirelessHART uses radios based on the IEEE 802.15.4-2006 standard. It operates in the 2.4Ghz band that is part of the Industrial, Scientific and Medical (ISM) bands. These bands are intended for unlicensed use. They are used by devices ranging from microwave ovens to wireless LANs and cordless phones. Hence, the probability of interference is relatively high.

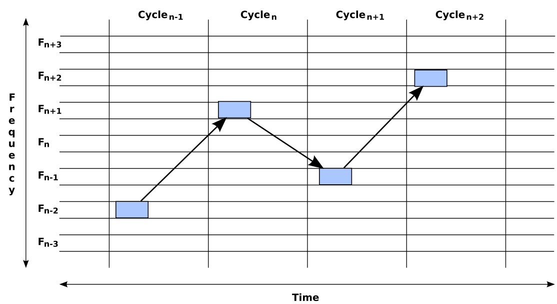

Figure 3.13: *Channel Hopping*

Data Link Layer To coexist in the 2.4 GHz band WirelessHART uses several mechanisms to minimize interference from other wireless systems. Time Division Multiple Access (TDMA) and channel hopping are used to control the access to shared medium. Wireless HART's TDMA protocol provides collision free, deterministic communication by the use of time slots. Time slots are pre-scheduled and have a fixed length (10ms). Precise time synchronization is crucial to the operation of the network. Time slots are synchronized between devices.

Each node synchronizes its internal clock with a selection of its neighbors. The selection of which nodes to use as clock references are done by the network manager (described below). It is the responsibility of the network manager to keep these nodes' clocks accurate. When a node receives a packet, its arrival time is recorded and compared to the ideal time at which the packet should have arrived. The difference between the two times are sent in every ACK destined for the source node.

To increase reliability, TDMA is used in combination with both channel hopping and Carrier Sense Multiple Access (CSMA). CSMA is used to verify the absence of other traffic before transmitting on the medium. The combination of TDMA and channel hopping enforces that devices communicating have to rendezvous in time and frequency, as shown in Figure 3.13. It is possible to ban channels which is useful for channels with high interference.

Network Layer and Transport Layer The basic building blocks of a WirelessHART network are:

- the *Field Devices* attached to the plant process
- a *Handheld* is a portable WirelessHART device used for maintenance task (i.e. running diagnostics, performing calibration and configuring devices)
- a *Gateways* connect field devices and host applications
- a *Network Manager* is responsible for configuring the network, managing of routing tables and scheduling communication between devices.

WirelessHART forms mesh networks with redundant paths between nodes so that packets can be routed around obstacles (e.g. dead links). All devices in the network must be capable of routing packets. The standard defines two mechanisms for routing that must be supported by all devices, namely graph routing and source routing.

A Graph Route is a list of paths that connects nodes in the network. The paths in each graph are configured by the Network Manager and stored at each device. A device routing a packet looks up the graph ID (stored in the packet's network header) and forwards it to one of the neighbors listed.

A Source Route is one path from the source device through the network to the destination device. The device-by-device route is specified in the packet and if one of the intermediate devices fail, the packet is lost.

Application Layer The application layer in WirelessHART is very similar to the one in HART since legacy HART devices must be compliant via wireless adapters.

Implementation Challenges We are currently not aware of an open-source implementation of WirelessHART. The WirelessHART standard is quite demanding in terms of performance constraints and memory requirements. Therefore it is challenging to provide an implementation for the most resource-constrained devices that adheres to the standard.

3.5.3.6 ISA SP100.11a

This standard is similar to WirelessHART but more general in that it will support multiple protocols via single wireless infrastructure. There is no open specification yet but it is assumed that the lower layers will be similar to the ones in the WirelessHART standard. In addition, the standard will incorporate 6LoWPAN.

3.5.3.7 Web Services on Devices

Recent attempts to adopt SOA paradigms at the device level inspired by development of more powerful electronic devices is expected to bring new opportunities to many business domains. Construction of SOA-based networks of interoperable resource-constrained embedded devices offering their functionality through services brings a new concept of "smart" devices, and premises new business solutions supported by integration of device functionality into business processes [225].

Devices Profile for Web Services Devices Profile for Web Services (DPWS) [67] is attempting to fully integrate devices with the web service world. DPWS defines a minimal set of implementation constraints to enable secure web service messaging, discovery, description, and eventing on resource-constrained devices. DPWS is an effort to bring web services on the embedded world taking into consideration its constrained resources. Several implementation of it exist in Java and C, while Microsoft has also included a DPWS implementation (WSDAPI) by default in Windows Vista and Windows Embedded CE.

The DPWS stack supports the following web service standards: WSDL 1.1, XML Schema, SOAP 1.2, WS-Addressing, WS-MetaDataExchange, WS-Transfer, WSPolicy, WS-Security, WS-Discovery and WS- Eventing. As a result, dynamic device and service discovery can be realized, while the metadata exchanged can provide detailed information about the devices and its functionality. This is well supported in DPWS with the inclusion of the main data discovery and transfer protocols such as WSDL, SOAP, WS-Transfer etc. Therefore, not only custom made device drivers can be eliminated to a large extend, but also these devices can now be easier and better used by enterprise resource planning applications via widely used technologies such as web services. DPWS has been tried out in a number of industry automation scenarios e.g. [225],[405] and device simulations e.g. [226, 227].

In August 2008, the OASIS Web Services Discovery and Web Services Devices Profile (WS-DD) Technical Committee was created to further advance the existing work, e.g.

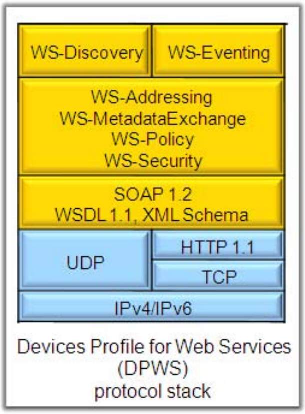

Figure 3.14: *DPWS stack*

DPWS. On May 13, 2009, the OASIS Web Services Discovery and Web Services Devices Profile (WS-DD) Technical Committee unanimously approved the following specifications as Committee Specifications:

- Web Services Dynamic Discovery (WS-Discovery) Version 1.1
- Devices Profile for Web Services Version 1.1
- SOAP-over-UDP Version 1.1

The specifications have now been submitted for consideration as OASIS standards. It is expected that the announcement of the submission will be made to the full OASIS membership on June 1, 2009, and the voting period will be June 15-30, 2009.

OPC Unified Architecture The OPC foundation actively develops the OPC Unified Architecture (OPC-UA), with the goal to advance the OPC communications model (namely COM/DCOM) towards service-oriented architectures and introduce a cross-platform architecture for process control. OPC-UA is a set of specifications applicable to manufacturing software in application areas such as field devices, control systems, manufacturing execution systems and enterprise resource planning systems. These systems are intended to exchange information and to use command and control for industrial processes. OPC

Unified Architecture defines a common infrastructure model to facilitate this information exchange.

REST Representational State Transfer (REST) is a software architectural style for distributed hypermedia systems like the World Wide Web. The term originated in a 2000 doctoral dissertation [127] and has quickly passed into widespread use in the networking community. REST is an architectural style for building large scale networked applications. REST describes a networked system in terms of data elements (resource, resource identifier, representation), connectors (client, server, cache, resolver, tunnel), and components (origin server, gateway, proxy, user agent).

3.6 Conclusions

As noted in this chapter, Cooperating Objects research puts together a series of highly dynamic and multi-disciplinary areas that cover aspects of both hardware and software, as well as their integration into functioning systems that work in the real world. Given the relative youth of the field, there is also need to perform research on the supporting tools that enable the programming, debugging and integration of such systems.

Although we have tried to cover as many aspects of the field as possible given the expertise of the authors, it seems clear that this overview of state of the art cannot contain all aspects of research. Nevertheless, we are confident that we have been able to select some of the most promising ones from the point of view of industry and academia. A more thorough analysis of research in terms of gaps will be the focus of chapter 6.

Chapter 4

Innovative Applications

Following a similar approach to the previous chapter, where we have looked at the state of the art in Cooperating Objects research, this chapter deals with the most promising applications whose innovation factor can be followed back to Cooperating Object technologies. The level of detail as well as the degree of visionary foresight depends on the amount of work industry has put into the different application areas. Therefore, some of them are more mature than others.

In this chapter, we concentrate on industrial, building and home automation, energy applications, transportation, environmental monitoring, healthcare and assisted living, and security applications.

4.1 Industrial, Building and Home Automation

Wireless sensor networks have gained much attention in the last years. Advantages of using wireless solutions for industrial applications are very often advertised:

- Ease the installation of I/O in difficult places
- Ease the design of modular machines
- Well adapted to moving machines
- Reduced installation cost (no network cable paths)
- Greater flexibility and so reduced modification time
- No downtime due to maintenance of network cables or connectors

The representative application areas for sensor networks for the industrial market are:

- Quality control with production process
- Machine condition monitoring
- Inventory tracking
- Monitoring of process parameters like pressure, flow etc.

The Building and Residential markets from thier side are adopting WSN solutions which can answer to the growing demand for comfort and security at home, the awareness about energy consumption, the strong needs for better energy management, the evolution of regulation, the demand/response programs and others.

The application areas for building and home markets are focusing strongly on building energy conservation systems:

- Adaptation of living environment to personal requirements
- Monitoring and control of humidity, heating etc.
- Monitoring and control of light using occupancy and activity sensors

A key promise of the wireless technology in building operation is to reduce the cost of installing data acquisition and control systems. With low-cost wireless sensor and control systems, not only will the cost of system installation be significantly reduced, but it will become economical to use more sensors, thereby establishing highly energy efficient building operations and demand responsiveness that will enhance our electric grid reliability.

4.1.1 Building Automation

Wireless technology has transformed communications in many areas, such as cell phones and PC networks. Now, with recent advances in technology and cost, as well as the emergence of standards, wireless solutions are ready to be deployed in building automation networks.

The cost of wiring alone is incentive enough for many building owners to look at wireless control systems, since wireless installations can be done anywhere, at any time, saving from 20% to 80% of the installation cost of controls.

Wireless building automation solutions are offering high flexibility, fast deployment, low installation costs, increased comfort and convenience. Wireless coverage enables not only wireless connectivity of personal devices, but also many other applications, such as automatic monitoring and control of all kinds of building management and security systems, motion detection and indoor positioning. Since wireless network technology provides greatly expanded data collection capabilities and carries inherent technical vulnerabilities, privacy and security of data are major issues to be considered within the scope of wireless building automation.

Much of the available automation technology, mostly developed first in industrial applications, would be useful in houses. The concept of intelligent building is not new; only the practical implementation has been very slow. Nowadays, no single technology or architecture could prove ideal across today‘s broad range of building architectures and occupant needs and many building managers are choosing hybrid (i.e., wireless combined with wired) solutions.

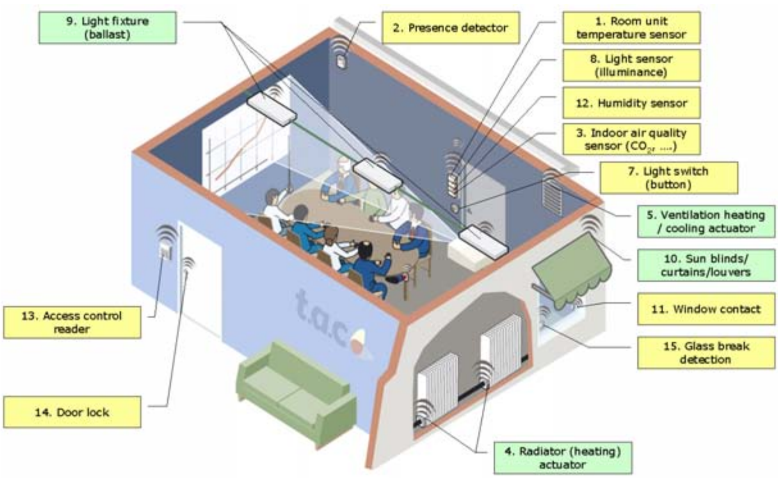

Figure 4.1: *Future Building Automation*

4.1.1.1 Business Drivers

One of the greatest benefits of wireless networks in building automation may well be energy efficiency, made possible by the easy, low-cost deployment of sensors and controls wherever needed. Usually, deploying more sensors and actuators means obtaining more energy-efficient buildings.

Accurate energy consumption monitoring of a building electric infrastructure, such as elevator, lighting, air conditioning, fire alarm system, ventilation, high and low voltage power distribution, etc., is one of the key issues to achieve an energy-saving or energy efficient building. In the construction of new buildings and updating existing building to install the energy consumption monitoring system to conserve energy, the most pressing issue is the high cost of integrated wiring and the high cost of repairing after update.

Therefore, for both new building and the existing building, the best way to transmit a message is through wireless means.

Today's commercial buildings are managed by large sets of sensors and actuators which control such functions as lighting, heating, air quality and security. Optimizing energy efficiency and increasing the comfort of occupants are two strong drivers for using complex building management systems.

Tertiary and commercial premises typically rely on standardized communication protocols (such as BACnet and LonWorks) to transport command and sensing data across building networks. To date, such communication networks have been almost exclusively deployed on wired media. Although the current cost of a wireless solution may not be always lower for new constructions, it is definitely a strong driver for retrofit installations or facilities in which it is highly expensive to install a wired communication network (e.g., buildings with concrete walls, museums, and architectural or historical sites that cannot be disturbed). In these scenarios, adding wired controls and sensors may end up being much more costly than deploying a wireless solution. Wireless chips can be embedded in devices like controllers, switches, and sensors for light, temperature, air quality, or presence detection.

Wireless also brings new flexibility to building control. Instead of placing controls where wiring permits, building owners are free to place controls where needed to improve building performance. This could have a major impact on energy efficiency, reducing wasted lighting and heating expenditure by 50 percent in many cases.

Currently, the residential and tertiary buildings are responsible for 40% of global energy consumption and 25% of CO_2 emission and the old buildings (built before 1975) – for approx. 70 % of total energy consumption. The EU objectives are to achieve 20% of energy saving by 2020 and having a directive on building energy performance. By enabling the collection of a much larger amount of data, wireless systems pledge even greater energy and cost savings associated with better optimization of lighting and HVAC (Heating, Ventilation, Air Conditioning) functions. The increasing pressure to improve building efficiency by even a few percentage points makes such solutions worthwhile.

Since energy-efficient control systems typically produce double-digit savings, the ability to go wireless for low installation costs is a powerful incentive for many organizations. By installing a wireless mesh control network over a wide area, many organizations can reduce wasted lighting and heating expenditure by 50 percent.

4.1.1.2 Next steps

With the problems of wireless networks resolved, and the emergence of standards now well underway, analysts expect new applications to be developed that take advantage of the unique capabilities of wireless technology.

The newest and most revolutionary form of wireless networking developed specifically for the building automation industry is the wireless mesh network, where short-range wireless devices and sensors self-organize into a network and can immediately reconfigure around a failure, if one unit in the mesh stops working. Mesh networks use distributed intelligence to communicate with all other devices within range. Not only can all nodes send and receive messages, but they also function as routers and can relay messages for their neighbors. Through this relaying process, a packet of wireless data finds its way to its ultimate destination, passing through whatever intermediate nodes are available. Instead of placing controllers where they are easy to wire, controllers can be placed where they are actually needed to optimize building performance. Wireless mesh technology promises to make building automation as common as traditional computer networks. In a few years one will be able to go down to the local Home Depot and pick up sensors and controlled devices, and quickly install and configure them. The next few years will be exciting as wireless mesh technology revolutionizes building automation and security.

The relentless push forward in semiconductor manufacturing processes will as well continue to drive down the cost of wireless networking devices while improving their functionality, creating an opportunity for more complex networking protocols that will improve system reliability, decrease system installation and provisioning costs, and create overall system flexibility unachievable with wired systems. To date, the adoption rate of wireless technology in building automation applications has been slow because of the unfamiliarity of the technology and the typical cautious and risk-averse mentality in this industry. However, with more successful technology demonstrations and falling cost of wireless networking technologies, wireless systems are poised to make significant inroads, particularly in the retrofit applications.

Areas that were impossible to wire and therefore not even considered are now not only possible, but practical and cost-effective. For example, one analyst states that the structural health of buildings will be monitored in new ways: Advancements in nanotechnology are enabling production of tiny sensors which can be placed at various joints, reinforcements, and other places during construction of a structure. These sensors constantly monitor the structural health and provide accurate data regarding cracks, excessive loads, or any other critical situation.

Meanwhile, wireless technology itself is expected to continue advancing. On the horizon are frequency-hopping technology to improve network connectivity, advanced security algorithms, and continued reductions in size, cost, and capabilities. In addition, the emergence of standards in wireless networking, such as ZigBee and BACnet, will help accelerate adoption by providing interoperability and a smooth transition path for building owners[51, 451].

4.1.2 Home Control

Wireless systems are becoming an important part of many new home area networks. They give high levels of portability and convenience to homeowners, and allowing a wide variety of remote devices to communicate with each other without expensive, complicated and awkward wiring layouts. In the home environment there are several home networks, such as: Internet access, computers and devices interconnection, audio and video distribution, security and surveillance system, energy saving systems, automation and control systems. Each one of these applications has band width, data rate, maintenance and installation requirements that are quite different and specific. A Wireless Sensor Network plays a very important part in these scenarios, being, usually, a solution with lower implementation costs.

The home environment automation and control systems, best know as home automation systems are developed with the intention to automate processes, like the illumination and remote control of the home environment equipment. The purpose of these systems is providing higher comfort and safety to the inhabitants [51].

The possible applications of a home network can be divided into main four groups: computing, entertainment, communications and automation. Even if automation home networks are still only a draft of the future usability, it is supposed that they will connect security, lighting, and heating systems together for the purposes of the user's convenience and energy management.

4.1.2.1 Business Drivers

Along with building automation, home control is expected to become one of the top market segments in terms of product deployment. The type of use foreseen for WSN in this area encompasses everything from domestic TV remote controls and central heating to lighting, rolling shutters, and alarm systems. Developing such devices requires great OEM involvement, which in turn needs large markets to generate adequate return on investment. Although vendor and product interoperability represents a strong confidence factor in gaining access to large markets, this is not the principal driver for adopting a standard in the home automation segment. Cost issues are much more sensitive than those in building automation systems. Beside cost issues, power consumption is also an important factor. Allowing customers to use easily reposition switches and other low-end products while getting rid of power mains is a strong driver for adopting the wireless technology.

Although batteries are not foreseen to disappear soon, scavenging energy to make control devices fully autonomous brings great prospects to both new construction and renovation markets.

Recently, wireless home-control products such as light switches, thermostats, blinds /drapes and appliance controls have reached the market. To have a true mass-market for these products, it is important to have a low-cost technology that is easy to install and operate. This requires a lightweight system that, from the end-user or installer perspective, is easy to install and requires no ongoing network management. The network must be a self-organized mesh network that ensures error-free communication and, in the case of malfunction, uses self-healing mechanisms to re-establish a reliable network. To support a full home-control system, the technology must support horizontal applications, enabling different product types from various vendors to communicate with each other and use each others' functionalities. To reach low cost points, the RF platform must be highly integrated, manufactured in low-cost processes and the associated software protocol must be lightweight.

4.1.2.2 Visions for the Future

A successful deployment roadmap for home control applications has to deal mainly with the following provisions:

- Low cost – To have a true mass-market technology, the physical wireless platform must be low-cost. The right trade-offs between technology choice and cost must be made without compromising the reliability of the network.
- Control protocol – The home-control protocol must address the required network traffic pattern while supporting network flexibility, reliability and ease-of-use. A home-control network is characterized by relatively few nodes (20 to 200) within a 150 m to 600 m area where each node communicates relatively infrequently (e.g. every 5 to 15 mins).
- Ease-of-use – The main challenge for easy network installation is to balance the requirements for easy network-joining and the requirements for easy identification of the installed devices. A number of different network-joining philosophies exist, ranging from full plug-and-play to manual processes with serial number typing. Most of these philosophies have shortcomings in real life due to the limited user interface on the typical home-control product with one or two actuators and indicators.
- Interoperability – A central challenge in product interoperability is to balance full interoperability with the vendor's requirement to be able to differentiate in the market. Furthermore, the interoperability requirement should reasonably match end-user expectations. The average user does not expect that all functionalities are identical in two products. However, he will expect that all basic functionality is the same or at least behaves logically. Interoperability is the basis for creating complete home-control systems in which different applications from different vendors work together. Product interoperability requires standardization on two levels: command level, where all commands that can be transferred between nodes must be standardized; device level, where all products must be a member of a device class that defines which of the commands are mandatory, recommended and optional.

Research is being conducted on case studies for the future on intelligent houses. The intelligent house is a network that interconnects several sensors and existent equipment in the house, as well as other networks that make available their services, in an attempt to provide the inhabitants a better safety and quality of life. One of the challenges of creating this services network is the need of addressing all the network nodes – for instance we can access a service in an external network. For an inhabitant or a group of inhabitants a profile can be created in the network. The individuals' presence and profile is identified in its home network, for instance, through the authentication of a Pocket PC, a mobile phone, or any other device. After the acknowledgment of the inhabitant's presence in the house, the environment can be adjusted to the inhabitant's profile, for instance, altering room temperature and brightness in each division he enters, accordingly to the inhabitant's preferences, season and time of day [191, 451].

4.1.3 Industrial Automation

In the last years the use of Wireless Sensor Networks in industrial automation has gained attention. WSNs are technically a challenging system, requiring expertise from several different disciplines.

In the industrial automation systems, there are numerous tasks to be considered, such as different means of supporting emergency actions, safe operation of the plant, automated regulatory and supervisory control, open loop control where a human being is a part of the loop, alerting and information logging, information uploading and/or downloading. Some of these tasks are more critical than others. The industrial automation systems are complex and often very expensive. In the future, Wireless Sensor Networks may be applied to realize cost effective and efficient automation with simpler mechanisms, which fulfill exactly the same functions as the existing problem solutions that have been in use.

4.1.3.1 Business Drivers

Industrial automation is a rather ambiguous market segment for wireless technologies. Although initially presented as one of its primary target industries, industrial control now tends to be perceived as a likely late adopter, given the specific constraints of manufacturing and process environments.

Robustness and real-time communication requirements are often cited as potential obstacles to using wireless technologies for industrial control. An even bigger impediment is the conservative mindset that characterizes the adoption of new technologies among industrial automation customers.

Given the large and diverse range of communication requirements in industrial environments, the deployment roadmap needs to be split into two major application categories:

1. Open-loop applications refer to typical monitoring and data collection tasks that are not part of the control loop itself. Wireless communications are used in these cases for noncritical operations such as commissioning, diagnostics, condition monitoring, etc.

2. Closed-loop applications are in contrast related to critical tasks since they are part of the control loop. Examples include real-time process control or safety-related operations like commanding valves. Such scenarios will likely not be part of wireless technology

roadmaps for 3 to 10 more years.

4.1.3.2 Industrial Applications Visions

It is expected that the use of sensor networks in industry will expand. At the moment there are only a few real applications, mainly due to the strict requirements for safe operations and criticality in operation, even if several demonstrators have been built. However, these issues will be studied and solutions will be provided in the near future. Another issue is time criticality, which is important in many control applications. The increasing bandwidth of the transferred data will help the development of these communicating platforms, especially for industrial use. Fast development of hardware, especially chips and their performance, will also provide increasing computation power, which will enable on-line analysis and decision making in the nodes of the distributed sensor network.

The first set of applications that will be ready for deployment in manufacturing and process environments are open-loop systems. Combined wired/wireless networks might be the first instantiation, in which control data are transported through wires, and a wireless link is used for commissioning and configuration purposes. Other promising areas include the monitoring of process variables, production equipment, or clamp-on types of instruments for permanent or temporary installations.

While acquiring sensing data is fairly easy, conveying these data to a central control system remains a significant challenge since it requires tight integration with existing automation infrastructures. If a large number of devices is to be deployed, power consumption may become a critical issue. Changing batteries frequently is either not feasible or not wanted by plant operators. This means that advanced energy optimization techniques need to be devised to maintain an acceptable service level while significantly decreasing duty cycles. Further down the road are closed-loop applications involving wireless communications. Several technical hurdles, such as guaranteeing high quality of service and implementing appropriate time synchronization mechanisms, need to be addressed before these scenarios become reality.

The overall data handling capacity will increase and history information – e.g. measurements – will be utilized in more detail. Power consumption, as already mentioned above, is another challenge for many applications but in the future there will be solutions to harvest energy for the whole lifetime of the network. Sensors will utilize and share common data in the network, not just analyze their own measurement. Plug-and-play and easy-to reconfigure operations will help the adjustment of the network for best and efficient use. The technology in industrial applications will be based on the common technology used in other applications, such as RFID, ZigBee, UWB, Bluetooth and IEEE 1451 standard for smart sensors. It is possible that there will be special versions of these technologies for industrial use that focus on the special requirements.

- Short-term steps to the vision: The current measurement systems that are mainly based on the point-to-point measurement topology will be replaced by simple networks. These networks will be able to utilize the measurements from all sensors and not crash if one or a few sensors or appliances are out of operation. The overall reliability of the networks will be improved as well as the speed of the communication.

- Long-term steps to the vision: In the long term there will be flexible sensor networks that can be either manually or automatically reconfigured locally or remotely. The sensors and appliances will generate their own power for the whole of their lifetime [417, 451].

4.1.4 Wireless Device Strategies

4.1.4.1 Discrete Manufacturing

Discrete manufacturers are keeping a close eye on development of wireless sensor standards in the process industries. But while the business drivers are in place, including wireless status as "the ultimate Fieldbus" the lag in development of wireless standards suitable for discrete applications will contribute to slower adoption at the device level.

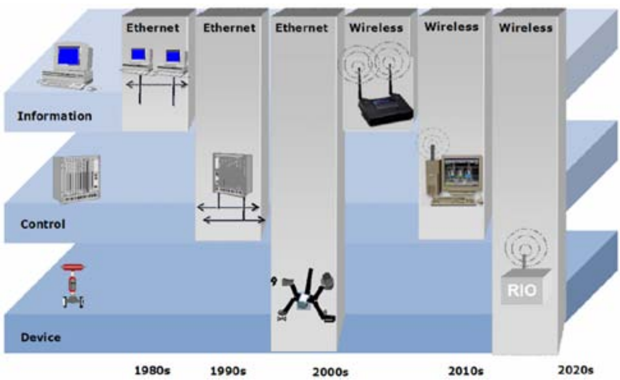

Figure 4.2: *Wireless migration to the device level*

Industrial Ethernet currently offers no wiring savings at the sensor/ actuator level, while wireless has the potential to be the ultimate fieldbus. This will lead to more rapid penetration of wireless at the device level once latency and performance issues are resolved and industrial standards are in place.

In the discrete sensor/actuator layer, all indications point to future wireless sensor interfaces that support IEEE 802.15.4-based physical layers and industrial protocols. These will incorporate some type of wireless power derived from magnetic or inductive energy harvesting and support functional or machine safety protocols for the machine building segment. This will ultimately result in a variety of wireless modules on the market that support specific industrial protocols and/or wireless functionality, similar to the variety of industrial Ethernet or device network components available today. The prospects for wireless adoption in the cable replacement segment of the discrete wireless market (the portion targeted at sensors, actuators, and remote I/O), are further out than those of the WLAN-based segment. Elimination of moving cables in rotating or mobile machinery, robots, AGVs, or harsh production environments is a primary target of the existing supply base in the area. Wireless sensors and actuators employed in these applications offer a significant value proposition, since cable failure is a leading cause of downtime. Despite the currently slow adoption rate, the prospects for wireless technology remain compelling.

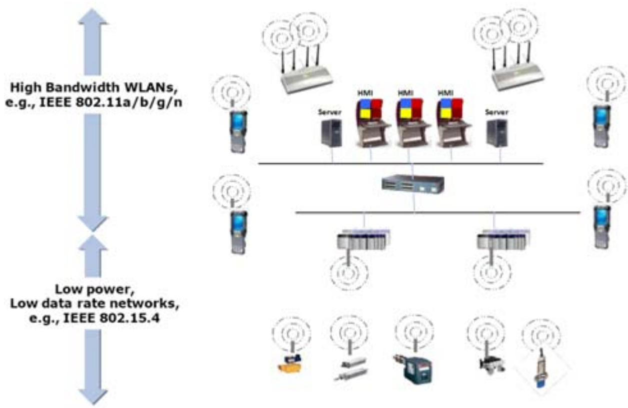

Figure 4.3: *Sensor interfaces will employ low power, low data rate networks*

However, issues such as higher speed discrete processes that cannot tolerate the latency of current wireless communications and the longer potential timeline for standardization at the discrete sensor/actuator level are just a few of the potential detractors to adoption. Power consumption and battery life have long plagued wireless sensor installations. This is even truer in industrial applications, where customers typically don't want to have to change batteries in remote or hazardous locations even once every five years. Ability to deliver wireless power enhances the value proposition of the sensor or actuator. This is due to the further elimination of power cables and a greater similarity to traditional device-level cabling that delivers power along with the measurement or control signal. The addition of functional or machine safety protocols is another increasingly possible scenario under the "industrialization" of the wireless interface.

Recommendations

Discrete manufacturers, who are monitoring wireless developments, recognize that mobile computing in the form of handheld computers and mobile HMIs will dominate the first wave of adoption, with wireless sensing and actuation further out. High-end industrial devices (such as mobile computers), will become the focus for wireless convergence as their current repertoire of WLAN and Bluetooth compatibility is expanded with GSM or 3G Cellular, RFID, location-based services, and higher-level industrial network protocols. Discrete manufacturers can achieve a strategic advantage, however, by preparing now for the inevitable time when wireless presents a more universal option for plant floor applications.

To this end, activities should include gaining an understanding of existing and potential technologies, identifying site-specific wireless requirements, participating in corporate RF spectrum management activities, and contributing to industry efforts in the areas of technical requirements, best practices, and use cases. Interim activities can also include installations using current proprietary protocols or evaluation applications in off-line or remote peripheral processes that have no real-time impact on the process. The move to wireless should start with local applications to minimize interference issues or conflicts with other wireless entities. Several industry sources, including the ISA 100 Factory Automation Working Group, are developing requirements documents that cover discrete and hybrid applications. Manufacturers typically augment these base requirements with their own application requirements as performance, security, reliability, latency, and similar current wireless inhibiting aspects. These requirements can then be compared to the specifications of emerging standards targeted at discrete manufacturing as they are released.

Forward-looking manufacturers are institutionalizing best practices in radio frequency (RF) spectrum management, often with the IT department responsible for managing at least the backbone portion of a company's wireless infrastructure. RF spectrum management is needed to ensure that the different internal silos pursuing wireless installations don't interfere with each other. Spectrum management can span issues ranging from sanctioned standards and suppliers, to internal channel assignments, known sources of interference, and responsible individuals for each. Manufacturers recognize that numerous potential sources of RF interference, as well as devices that could be subject to RF interference, reside in and around their facilities. In general, the wireless industry standards organizations have improved the RF frequency management outlook though their own better management practices, i.e., reducing the possibility that devices supporting different IEEE standards will interfere with each other.

Since, in most facilities, manufacturing is not the lead player relative to wireless adoption, it is important to incorporate these spectrum management decisions into plant floor installations. IT-dominated spectrum management strategies can lead to internal opposition to the possibly unfamiliar networks likely to be used at the device level of discrete manufacturing, including Bluetooth and IEEE 802.15.4.

While emerging wireless standards may offer the promise of interoperability among differing vendors products, most discrete manufacturers are looking to standards to address performance issues in areas such as latency and coexistence with other networks [157].

4.1.4.2 Process Manufacturing

There are a number of important factors that make wireless the most important emerging technology today in manufacturing automation. Manufacturers, even risk-averse process manufacturers, recognize that wireless can offer cost reductions. More importantly, leading manufacturers (again especially in the process industries) see wireless technologies as an enabler of entirely new business processes that will not only be less expensive, but will be safer, more reliable, and far more transparent than their current manufacturing practices.

The industrial wireless market is growing as newer wireless technologies are introduced for new applications. In general the shorter the range, the newer (and less mature) the industrial wireless technology.

The oldest and most traditional wireless manufacturing applications are in the process industries. Here wireless communication serves as one of several options in SCADA systems serving the oil and gas, power, water, and wastewater utility industries. Wireless is chosen as an alternative to other SCADA communication technologies such as fiber optics, leased telecom lines, or private networks, applications that use wide-area network (WAN) technology. Longer range wireless for SCADA applications still represents a major share of the industrial wireless market, but not likely to be one that will experience the most growth. However low-cost wireless field devices such as pressure transmitters, temperature transmitters, flow meters and computers will help to expand the use of wireless technologies in metering and natural gas compression applications. These newer applications will complement the SCADA applications and will likely increase wireless SCADA communication adoption since the entire system.

The new and growing applications do not use WAN, but instead use local area networks (LANs) and to a lesser degree wireless sensor networks. WLAN, now part of the IT mainstream, is moving more and more into manufacturing applications, primarily in discrete manufacturing. Suppliers of manufacturing automation are feverishly trying to fit both WSN and WLAN into their products.

The most important factor favoring wider deployment of wireless technologies in manufacturing is the rapid rate of development of various wireless technologies in today's market. Manufacturing applications of wireless can to a fair degree ride this wave of development.

Automation companies have become increasingly comfortable with outsourcing non-critical product components over the past two to three decades. Just as automation companies have outsourced wired networking embedded within their products to commercial suppliers, new products embedding wireless technologies can be developed with significant outsourced content.

A major factor favoring greater deployment of wireless technologies in manufacturing is the ability of wireless applications to enable new and better ways of operating manufacturing plants. While no sector of manufacturing is excluded from the impact of more pervasive wireless, process manufacturing certainly stands to feel the greatest impact since the technologies involved in process manufacturing change only incrementally, and the advent of wireless will drive changes in several areas simultaneously.

Though process automation technology has been very slow to change, process manufacturers continue to face external pressures that force their operations into greater complexity and interdependence than when their automation strategies were developed. Changes in feedstock and changes in product production runs are far more common than a few years ago. Enterprise procurement and enterprise sales and distribution applications can squeeze production plants from both ends with demands for greater flexibility. These types of operations require higher levels of automation, but they also require more collaborative processes in operating and maintaining plant assets.

In the process and discrete industries, the adoption of wireless enabled computers have proven their benefits in the maintenance of plant assets including machinery, production equipment and sensors and field devices that automate the control processes and production lines. These mobile solutions can make each worker an empowered expert regardless of experience [156].

4.1.4.3 New Wireless-enabled Applications

Difficulties of managing simultaneous production and maintenance along with more complex scheduling and increasing regulatory requirements for environmental performance and personnel safety drive process manufacturers to look at many more fundamental automation and monitoring technologies that can execute continuously in the background and provide an information infrastructure that gives digital visibility to plant events that now remain undetected and unrecorded. Examples of this new visibility include items such as the condition of process sensing lines, sensor diagnostics, wider use of redundant sensors, real-time tracking of the location of all personnel, continuous monitoring of more and more components such as gas cylinders, safety equipment and manual valves. Finally workflow processes enabled, monitored, and validated by wireless field communications will become part of the daily routine at leading process manufacturers in a few years.

Increasing reliance on service-oriented architecture (SOA) is designed to address this situation within enterprise applications. However in terms of field operations these activities require wireless devices, wireless monitoring, and wireless data entry which until recently has been too impractical to consider. Another force that will certainly drive wireless growth during the next few years is the emergence of new and valuable applications that are enabled by new wireless technologies. Unified voice and messaging communications is presently the leading such application. This is enabled by the convergence of voice telephony and IP networking. Over a period of only about 10 years virtually all of the

PBX equipment purchased by enterprises will transition from what are now Legacy TDM products to telephone switching equipment based on IP data streams. Such technology falls under the umbrella term voice over Internet protocol (VOIP). As voice telephony and voice messages become simply another IP data stream, these can be carried over a wide variety of networks both wired and wireless [156].

4.2 Energy

We have entered a new energy era where the world's economic regions are dependent on each other for ensuring energy security and stable economic conditions. Europe and the rest of the world share common aims of providing abundant clean, secure and affordable energy. Therefore, future goals include improving energy efficiency, increasing penetration of renewable energies, diversifying and decentralizing Europe's energy mix and enhancing competitiveness of European industry. In parallel, Europe's key strategic goal is to become the most advanced knowledge-based society in the world, including an environment for successful and exploitable research. The energy field and the increasingly closely related information and communications technology (ICT) domain offer a unique chance for European citizens and businesses.

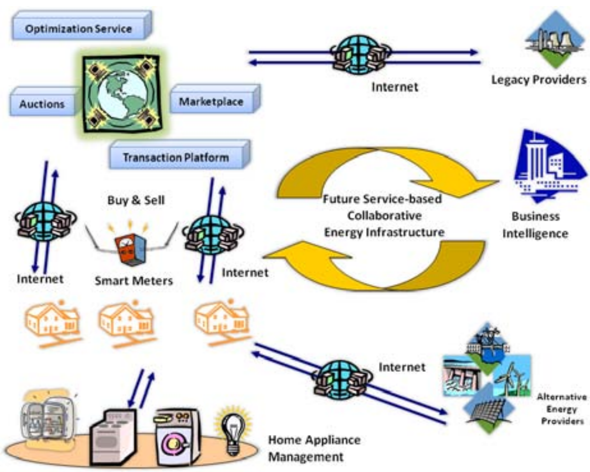

Figure 4.4: *The Future Energy Infrastructure*

Deregulation of the European energy market is seen as instrumental in achieving these goals. The intention is to establish a free and competitive market for energy production and consumer-integrated distribution by breaking up the value chain - production, transfer and distribution of electrical power. A much more decentralized and diversified production and distribution system will emerge. New technologies for co-generated heat and power and increased use of renewables such as biomass, solar energy and wind power will introduce a considerable number of diversified systems into the power grid, in addition to traditional large scale power plants. Consequently, the share of decentralized power generation - by industrial or private producers - will increase and have a dominating effect on existing infrastructure, technologies and business practices.

This paradigm shift will reshape the energy map of Europe as well as the associated business domains. Innovative new technologies and concepts will emerge as we move towards a more dynamic, service-based, market-driven infrastructure, where energy efficiency and savings can be better addressed though interactive distribution networks. A fully liberalized market will advance legacy processes, improve energy sustainability and security, create new business opportunities and have a positive impact on the everyday life of citizens.

In the future service-based Internet of energy [228], several alternative energy providers, legacy providers and households are interconnected. Via smart meters, one is able to interact with a service based infrastructure and perform actions such as selling and buying electricity. More advanced services are envisaged that will take advantage of the near real-time information flows among all participants. Furthermore the energy consuming/producing devices will be no more considered as black-boxes but will also get interconnected, which will provide fine-grained info e.g. energy optimization per device. Existing efforts in the emerging Internet of Things and Internet of Services, will be combined and be a crucial part in the envisioned Internet of Energy (Figure 4.5).

Power engineering alone, however, will not be able to transform the energy markets. New, highly distributed business processes will need to be established to accommodate these market evolutions and fully integrate the distributed electricity sources. The traditionally static customer process will increasingly be superseded by a very dynamic, decentralized and market-oriented process where a growing number of providers and consumers interact [228]. A new generation of fully interactive ICT infrastructure has to be developed to support the optimal exploitation of these changed, complex business processes and to enable the efficient functioning of the deregulated energy market for the benefit of European citizens and businesses. The architecture of such distributed system landscapes must be designed and validated, standards need to be created and widely supported, and comprehensive and reliable IT applications will need to be implemented

Future distributed energy systems should be robust, self-managed, self-sustained and enable dynamic reorganization and coordination of services and markets. Therefore the Internet-based infrastructure should be tightly linked to the energy domain, and be used to support the development of new mechanisms for coordinating real-time demand and

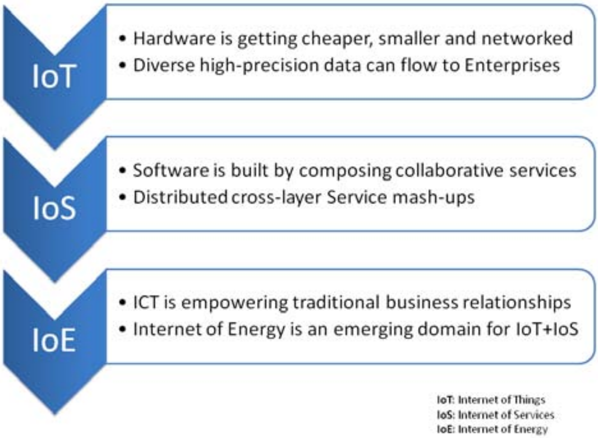

Figure 4.5: *Internet of Energy: Combining IoT and IoS*

responses in the electricity market - i.e. the consumption and feeding-in of power and the resulting interlinked commercial transactions. Transaction platforms will serve as electronic marketplaces facilitating the commercial activity associated with the purchase and sale of electricity and its derivatives, not only for utility companies but also for decentralized consumers and producers. Intelligent, interactive energy-management systems will be needed for an infrastructure capable of supporting the deregulated energy market. Intelligent electronic meters [229] will have to be installed for millions of households and companies, connected to the future transaction platforms. These 'e-meters' will be able to provide almost real-time data that in turn will have an significant impact on existing and future energy management models since they can now be based on real-world and real-time data. Policy and decision makers will have fast access to the data necessary for effective market regulation and competitive business models. Households and companies will be able to react to fluctuations in the market by increasing or decreasing consumption or production, thus directly contributing to increased energy efficiency.

Large-scale research efforts will have to deal with the opportunities and challenges associated with goal of closely relating ICT and the energy domain. They will entail the development of an appropriate security, safety and risk concept and architecture for an electronically-based energy market. In addition, an interoperability framework will need

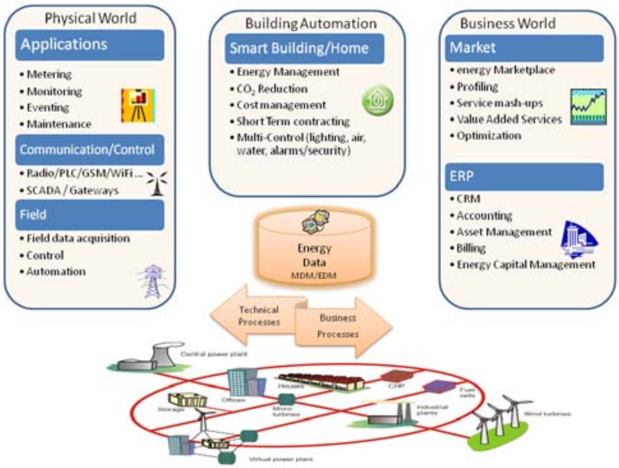

Figure 4.6: *Cross-Domain Cooperation for the Future Internet of Energy*

to be developed to enable the interoperation of the abundance of interfaces and systems that will inevitably result from a highly decentralized electronically-based energy market. Apart from robustness, resilience, trust and security, extended support for networked and managed businesses and service convergence across a multiplicity of environments should be achieved.

Today, we are at the dawn of an era that we would describe as the 'Internet of Energy'. Existing efforts in the energy domain can be compared to the Internet-based ICT efforts of the 1970s and early 1980s. Once a highly distributed and service-based interactive energy infrastructure is in place, we will see new innovative concepts and technologies flourishing that will empower us with new capabilities and tools to attack old problems. However, many challenges lie ahead in the next ten years.

4.2.1 The Smart Meter

We are moving towards the "Internet of things" [131], where almost all devices will be interconnected and able to interact. The same will hold true for energy metering devices. These smart meters will be multi-utility ones, managing not only electricity but also gas, heat etc. New information-dependent intelligent energy management systems will be needed for an infrastructure capable of supporting the deregulated energy market. Smart meters will have to be installed for millions of households and companies and get connected to transaction platforms.

Smart meters [229] provide new opportunities and challenges in networked embedded system design and electronics integration. They will be able not only to provide (near) real-time data but also process them and take decisions based on their capabilities and collaboration with external services. That in turn will have a significant impact on existing and future energy management models. Decision and policy makers will be able to base their actions on real-world, real-time data and not simple predictions. Households and companies will be able to react to market fluctuations by increasing or decreasing consumption or production, thus directly contributing to increased energy efficiency.

In the longer term, smart meters could even be the gateway of communication of household devices with the Internet. It is expected that smart meters will have advanced local communication capabilities (e.g. Bluetooth, IrDA, ZigBee, Wibree etc) and an Internet connection (e.g. via WiFi, DSL, UMTS etc). Therefore they could both participate in local ad-hoc networks with other household devices and in parallel be their communication medium with the outside world. This in turn opens up some interesting issues to be researched as well as the possibility to apply new business models. The replacement of legacy meters will not be linear depending only on the cost or energy provider's intentions, but rather a dynamic one. How fast we will move towards a fully fledged advanced metering infrastructure will depend on the co-evolution of technology and business opportunities.

In the near future, meters will transform themselves to embedded devices with CPU, memory, and will have the capability to execute general purpose code that implements third party services. Seeing the meter as a device with computing capabilities, allows us to define a layered open architecture for smart meters as the one depicted in Figure 4.7.

As seen, we have several layers that communicate with each other via APIs. These APIs need to be defined and standardized in order to allow for interoperable interaction.

- Programmable hardware: This is the lowest layer of the architecture e.g. the electricity meter and the basic software delivered by the manufacturer. In order to ease the integration of the hardware in other systems, the functionality offered has to be captured by the open hardware API. Also via the same API one is able to manage the hardware device i.e. program or configure it according to the capabilities offered.

- Embedded middleware: This layer is a general purpose middleware for embedded devices. Its role is to provide the capabilities for creation and support of execution environments (EE). The middleware manages the lifecycle of the EEs and is able to also capture the hardware's capabilities and offer via a multitude of APIs a finer programming environment to the EEs.

- Execution Environments: The execution environments (EE) are hosted by the middleware and provide specific capabilities that service providers can use to deploy their

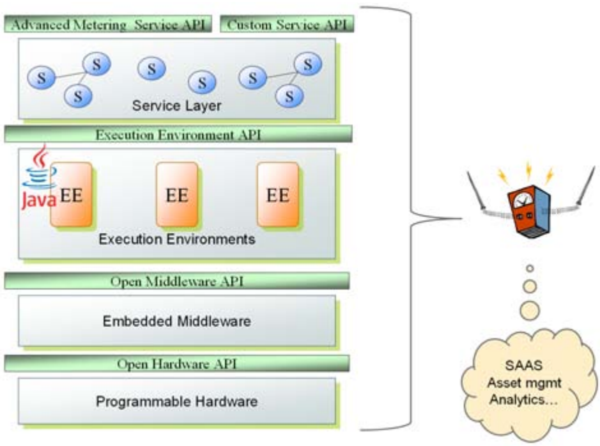

Figure 4.7: *The Smart Meter as a networked embedded device*

services. Each meter is expected to host at least one EE.

- Service Layer: Several services run in the different EEs on the mete-ring device and offer a standard API to the applications. One service can be standalone or depend on others to provide its functionality. The API offered by the services is standardized and is a uniform way of accessing, the capabilities of meters and programming it.

The main motivation behind this modular approach is that each layer should be agnostic of the other layers and only depend on the specific API below it. In a heterogeneous infrastructure such as that on future energy networks, many programming languages and a plethora of implementations are expected to exist for various reasons e.g. performance, flexibility, advanced capabilities etc. However, as long as the basic standardized APIs are globally implemented, all will have a common basis which will enable their interoperability. This is expected to ease also vertical integration at customer side that may be necessary to create robust and highly distributed deployments. Furthermore the existence of an execution environment implies that the meter can adapt its behavior and be incrementally software-upgraded.

4.2.2 Smart Meter collaboration

The existence of smart meters that can be also accessed in a seamless and uniform way via standardized methods is a must for the future service oriented infrastructure. Assuming that smart meters will be accessed e.g. via web-services, has far reaching implications, since now business processes can actively integrate them in their execution. Smart meters can provide real-time data which can then be consumed by services, which in their turn now can act based on rapid changing context conditions Furthermore, instead of providing only their data (one-way communication), which limited their usage up to now, they can now be active and host business intelligence (bidirectional communication) which does not have to rely only on the back-end systems.

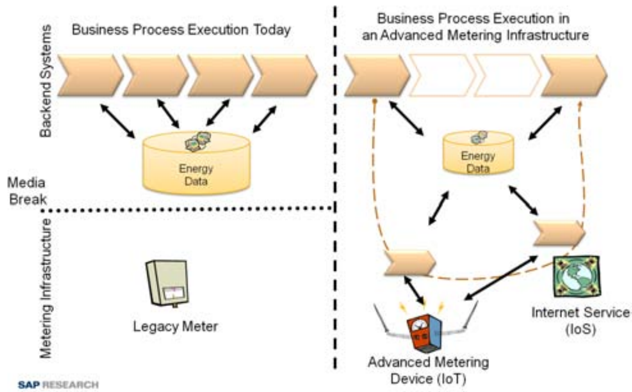

Figure 4.8: *Advanced Smart Meters in a collaborative infrastructure*

As depicted on the left side of Figure 4.8, a typical business process executes in a backend system that hosts the business intelligence. At some point the meter is interrogated by the business process and its metering status is sent in a time-frame which may vary greatly since the acquisition can be done electronically or even per post. On the left side of Figure 4.8 a similar process is depicted, which however assumes an advanced metering infrastructure in place. Since the meters do have computing capabilities, are able to process locally their data and take local decisions, this data does not need to be sent to the backend systems. Therefore we have a part of the business process executed outside the backend system. The business process could be even more distributed since the meter may trigger an external Internet service which will do advance the business process itself. So from the original steps (four are depicted as an example in Fig. 4) in the business process execution only two of them have been done at the backend infrastructure while two others have been executed collaboratively by the meter itself and another Internet based service. The advantages are profound i.e. more lightweight business processes which can outsource or parallelize specific execution steps, we have reduced communication overhead since the data do not have to be transferred to the backend but stay at their original source, and we are able to realize more sophisticated business processes that are highly distributed and may even partially belong to different domains. In an infrastructure where real-time data is constantly generated, needs to be processed and is composed of millions of devices (only in Europe there are more than 225 Million electricity meters), centralized processing (e.g. on backend systems) could be problematic, but such delegation of tasks and distribution of business intelligence may be another step towards more viable and better managed infrastructure.

4.2.3 Research Challenges and directions

The formation of new relationships between energy providers, distributors, dealers, and customers, who, themselves, can act as producers, has dramatically increased the complexity of the energy market. This increased complexity requires innovative IT solutions. Building on the experience gathered from previous research, new concepts and approaches must be developed, implemented by prototype, and tested in practice, the goal being to create an IT infrastructure for the deregulated energy market. Efforts should focus on finding realistic/viable approaches where the interests of those players involved are considered in addition to fielding purely technical questions. Authorities, utility companies, operating companies, software providers, providers of measurement systems, and power engineering providers must explore new cooperation models in a deregulated market.

Metering Infrastructure will evolve from the existing automated metering approaches that restrict themselves only to billing towards a diverse multi-utility infrastructure as depicted in Figure 4.9. There the billing will be expanded, to include also mobile meters (e.g. electric cars), while several value added services will be created.

An overall system architecture to assist the deregulated power market, especially decentralized energy systems will have to be designed and implemented. The core business processes covering all aspects e.g. (consumption and generation of electricity, derivative trade, requirements and production planning, maintenance of plants, etc.) will have to be adapted. Integration of embedded systems (e.g. meters and sensors) as well as the communication capabilities in the envisioned infrastructure need to be enhanced. modeling highly distributed business processes, developing market-driven mechanisms for load balancing, proactive planning of system load profiles using derivatives, development of new business and market models, allowance for planning and scheduling, and assurance of interoperability are just a few other topics that need to be specially defined and developed in this area.

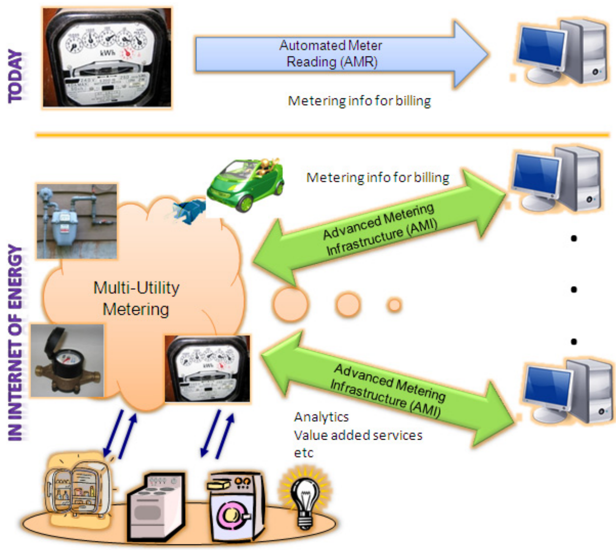

Figure 4.9: *The future of metering: Multi-utility value added services*

Generally research efforts will have to deal with the opportunities and challenges associated with the goal of closely linking ICT and energy. Development of an appropriate security, safety and risk concept and architecture for an electronically-based energy market will be the core. In addition, an interoperability framework will need to be developed to enable the interoperation of the abundance of interfaces and systems that will inevitably result from a highly decentralized electronically-based energy market. Service architectures, platforms, methods and tools focusing on a network-centered approach will need to be developed to support the networked enterprise. Understanding and managing the complexity of a critical infrastructure such as the energy sector is crucial and implies systemic risk analysis, resilient distributed information and process control frameworks.

From the business and application perspective a number of important aspects also need to be resolved. Directions for the short term should focus on: Transaction Platforms: One central question involves the consumption and generation of electricity and the interlinked commercial transactions. The object of research here is the business and legal identification of new market forms that are compatible with the revised general conditions. The direction the deregulated energy market will take in the years ahead is uncertain, but will be shaped to an extent by the behavior of today's market participants. Therefore there is industrial motivation to explore the different scenarios and not to leave the market structure of the future to chance, but instead actively shape it.

Concrete directions that need to be investigated and where cooperation issues arise include:

- Linking Decentralized Meters & Control Units: There is need for new, electronic meters and control units to provide an infrastructure capable of supporting the deregulated energy market. Smart meters are a special form of embedded system featuring a central unit, on-board memory, and diverse communication options. The unit can also create an energy profile of the household or company and is able to react to fluctuations in the market by intelligently increasing or decreasing consumption or production. This vision can even be taken one step further by also equipping each end device (motors and machines of all types) with sensors that calculate power requirements and forward this information back to the smart meter and the associated service infrastructure. Such system can take into account production and capacity planning in order to keep operation as cost efficient as possible. The goal is to create a link between these meters and the above-mentioned transaction platform for the energy market. This link must be feasible for millions of households on a decentralized basis, thus necessitating easy and reliable installation (p̈lug & trade)̈.

- Linking Production Planning: Most energy consumers are not aware of their daily energy consumption, and it is also not common practice to plan for energy requirements. As such, utility companies generally are not in a position to predict the needs of their customers and have no choice but to provide quick-access overcapacities, which in turn lead to expensive energy production. Energy requirements calculated using production planning can then be checked against the power market to optimize energy costs. With the help of derivatives (options) for planned power requirements, planning can be synchronized with the power market via proven mechanisms. These mechanisms can, in turn, play an increasingly important role, especially in the context of rising energy costs. Data formats and interfaces must be standardized to allow costs and energy consumption to be minimized through operative planning.

- Infrastructure Security: Security is critical for an electronically-based energy market to succeed. Secure methods for exchanging data between decentralized measuring points and the utility companies and customers using open networks such as the Internet must be in place. As an electronically-based energy market increases the flow of private information, steps must be taken to ensure that solutions protect personal data from being misused. In addition to scalability and robustness, a sufficient security concept is the key non-functional factor in designing an electronically-based energy market. Here, consideration also must be paid to the stipulations of joint European security and reliability standards.

- Linking Energy Management Systems: Future energy management systems (EMS) will be linked via wired or wireless-based systems for recording and regulating the energy flows of a consumer or producer. One particular problem, however, is how to deal with expensive peaks in consumption. In a more general sense, the objective of an energy management system is that of acquiring, using, and producing energy in the most efficient way possible. Generally, an energy management system is used in a commercial or industrial application, but there is good reason to believe that a lighter EMS will also applicable in consumer pools and even in individual customer dwellings. Linking such systems to the electronic energy market can introduce entirely new options for increasing the overall efficiency of energy grids.

- Interoperability: The highly distributed system architecture of an electronically-based energy market gives rise to an abundance of interfaces between different systems. The interoperability of all of these heterogeneous systems presents a decisive challenge to the functionality of the overall system. First, data formats need to be specified. Agreeing on such conventions not only necessitates the use of structures, or syntax; semantics also play a critical role. State-of-the-art IT concepts such as ontology and decision algorithms can be used as well. The next step involves the design of a system-wide process. Here, the appropriate methods researched on interoperability as it applies to corporate information systems, will be further developed for use in conjunction with electronic energy markets.

4.3 Transportation

4.3.1 Aerial Transportation

In aerial transportation domain, commercial aviation includes transport of both goods and people by airliners. Adoption of technologies in commercial aviation typically takes longer time than in other fields due to the strict safety requirements. However we can say that today computerization and digital technologies have become pervasive in all relevant fields for aerial transportation: commercial aircraft production, operations, and maintenance. The main change expected in the future is the architectural evolution from a set of disconnected systems interacting in a one to one basis to a system of systems designed from the beginning with the aim of cooperation among all systems. Cooperating Objectstechnologies are a key enabler for this new approach. A revealing sentence from [424]: "...in the new digital environment the aircraft is nothing but a mobile node in a world wide network infrastructure of various communities where every one's needs can be streamlined to provide the right data to the right end-user community at the right time to make the right decision. Some of the main issues expected are those related to standardization and certification."

This new future architecture is usually referred to as Network Centric Operations (NCO) or Net Enabled Operations (NEO). Examples of systems affected by this vision are: integrated aircraft systems, Satcom and Broadband satellite systems, Ground receiver systems, Factory/Productions Systems, Flight Test Systems, Flight Line Systems, aircraft Operations and Maintenance, Air Traffic Management (ATM), on-board wired and wireless systems.

NCO concept relies in the network capabilities to enhance the overall system performance while reducing operational costs.

Vision for commercial aviation can be summarized this way: within the next decade or two every commercial airplane irrespective of its location will be a securely and seamlessly connected node in the world wide web with Net Centric Capabilities enabled for all of its user communities around the world [424]. Moreover, each of this nodes itself will be a network of hundreds of sensors and actuators controlled by systems with different levels of autonomy, ranging from human operated ones to completely autonomous ones. This architecture fits with the hierarchical definition of Cooperating Object, formed by a group of other Cooperating Objects.

Number of sensors on-board has been increasing very fast in the last years, and it is expected than with cost reduction, miniaturization (weight reduction) and wireless capabilities the number will increase even faster, allowing the systems to monitoring in a more detailed way the status of the aircraft and increasing the safety (using better integrated health management) and the performance of the aircraft.

Flight Data Recorders (FDR) have been in place for a long time now and have evolved from magnetic tape based ones, collecting several tens of parameters, to current ones, based on solid-state memory board capable of collecting several hundreds of parameters.

One of the best know and older utilities of FDR is aircraft accident investigation, however due to the wide kind of data they can collect now they are becoming invaluable tools for other applications like maintenance and training.

From a safety point of view, Flights Safety Foundation (FSF) claimed in May 2004: "U.S. Must Mandate Airline Flight Data Collection and Analysis. This proven tool for improving aviation safety must be required by the government to be successful." FSF refers to flight operational quality assurance (FOQA) as a key instrument for safety, being basic for this technique the monitoring and analysis of Flight Data.

The Federal Aviation Administration (FAA) requires that commercial airlines record a minimum of 11 to 29 parameters, depending on the size of the aircraft. On July 17, 1997, the FAA issued a Code of Federal Regulations that requires the recording of at least 88 parameters on aircraft manufactured after August 19, 2002.

Thanks to knew development in sensors now it is possible to record movements, positions, temperatures, pressures, deformations, etc. with more precision then ever.

Some examples of the parameter recorded:

- Time
- Speeds (air, ground, true. . .)
- Acceleration in 3 axis
- Heading, pitch, yaw
- Flight control position (pilot command)
- Surfaces control position (angle)
- Fuel consumption
- Engine performance

The development of low energy, low cost, & energy harvesting wireless sensors will allow to install them not only in new aircrafts but in current ones while retrofit operations. Current efforts like Installed Intra-Aircraft Wireless Communications at ITU-R will have reserved spectrum for the onboard wireless connectivity.

The connection of the aircraft to the ground systems is another area which requires a lot of research in order to find a solution which provides a highly reliable broadband connection cheap enough for don't limit the data to the critical one for safety (traffic, weather. . .) but to include a wide range of services for the passengers [305]. Current limitation to voice services and a few Internet connections will be surpassed and new services like access to video on demand, virtual office, telemedicine and so on would be possible.

As in other fields off application of Cooperating Objects, interoperability issues will arise and should be managed to ensure smooth transition as aircrafts cross different regions of the earth.

From the maintenance point of view, current initiatives showing how the future can profit from net enabling and Cooperating Objects are:

Electronic Flight Bag (EFB) replaces volumes of paper documents and checklists that pilots have traditionally carried to the flight deck. It allows airlines to update and distribute documents, instructions, routing information and crew assignments electronically. Pilots use EFB to calculate takeoff and payload performance on the runway, giving airlines flexibility to make the most of the airplane's revenue capacity and fuel economy. Of greatest interest to financiers, EFB captures a digital record of pilot fault reports and details of the airplane's maintenance and operational history. Linked to a central maintenance database, this information can help owners make a complete and up-to-date assessment of the airplane's maintenance condition.

Airplane Health Management (AHM) monitors the airplane's condition in flight, providing real-time decision support to airline operations and maintenance personnel on the ground. AHM transforms cryptic fault codes and performance data into prognostics information that allows airlines to resolve maintenance issues before they cause major schedule delays or flight cancellations. The system prioritizes fault reports so airlines can schedule service efficiently. AHM's fleet-wide statistical analysis and prognostication ability help avoid unscheduled maintenance, which increases aircraft utilization. For financiers, this can mean fewer outstanding maintenance issues when the time comes to evaluate the airplane.

Integrated Materials Management (IMM) transfers much of the responsibility and cost for managing spare parts and logistics from the airlines to the manufacturer and its global team of logistics partners. The IMM team retains ownership of spare parts inventories so airlines pay for parts only when the parts are delivered to the work site for installation on the airplane. Parts are deployed near airline maintenance operations for prompt delivery directly to the work site. The manufacturer manages logistics, warranty, order tracking, and inventory replenishment. For airlines, this means lower acquisition costs and a streamlined supply chain. For financiers, reducing the initial spares and logistics component of acquisition cost means that the airplane's asset value can cover a larger percentage of the initial investment. And, because parts and delivery processes are standardized, the maintenance condition is more transparent and predictable.

Talking about the Supply Chain, parts tracking can benefit from enhancement in technologies about location identification, ubiquitous network connectivity, security, high performance low cost computing, decision tools, and open standards.

At operations Air Traffic Management (ATM) system is recognized as an obsolete system based in legacy technology that needs to be revamped to cope with future traffic increase (up to 3 times current traffic by 2020, see Advisory Council for Aeronautics Research in Europe (ACARE) Strategic Research Agenda (SRA) Goal [6]), a concept reflecting this new network based vision is SWIM (System Wide Information Management). Current ATM systems and operations are not network-enabled and are characterized by rigidly configured systems (communications lines, computers, and software applications), hardwired to geographically disparate facilities. Due to these limitations, current systems offer limited redundancy and security and severely restrict data sharing and interoperability. As a result, the current ATM system architecture is overly expensive, needed modifications are extremely costly and time consuming, and network-enabled operational capabilities needed to meet future capacity demands are not feasible. SWIM is an information management architecture for the future ATM system.

In the ICAO Global Air Traffic Management Operational Concept SWIM is described as thus:

"System Wide Information management aims at integrating the ATM network in the information sense, not just in the systems sense. The fundamental change of paradigm forms the basis for the migration from the one-to-one message exchange concept of the past to the many-to-many information distribution model of the future, that is geographically dispersed sources collaboratively updating the same piece of information, with many geographically dispersed destinations needing to gain situational awareness with regard to changes in that piece of information.

Successfully managing the quality, integrity and accessibility of this complex, growing web of distributed, fast changing, shared ATM information, called the virtual information pool, can be considered as the main operational enabler for the operational concept.

SWIM concept will benefit form increasing level of autonomy in the systems interconnected so they can cooperate in a transparent way to the final user.

Airports are a wonderful place to reproduce the vision of Mark Weiser; in a relative small place you can find hundreds computerized systems and intelligent devices. All users' communities; airliners (on-ground and flying staff), passengers, handling service providers, safety and security forces... have the need to coordinate among themselves. Connectivity at an airport is a key component that is crucial to the success of airlines. The key technical issues are High Band Width, Seamless, Secure Connectivity and Open standards. For example refuel operations could be streamlined by communication among the arriving aircrafts and the refueling services. Similar scenario is applicable for catering or de-icing services. Security services in the airport will know exactly the position of all assets and could detect strange patterns to pro-actively avoid incidents. Processes like check-in could rely on mobile devices ported by passengers and airliners would communicate one to one for each flight passenger on the airport; luggage could be tracked by RFID and detection of dangerous content will be automatically detected and treated. In the future WiMax or better connectivity will allow complete collaborative environment at the airport. Hundreds of devices with available computing capability could collaborate in grid style architecture to use this computational power as needed. All this advances will help to achieve ACARE SRA Goal [6]: "To reduce the time spent by passengers in airports to under 15 minutes for short-haul flights and to under 30 minutes for longhaul".

Transport of goods (freighters) have the special characteristic of having much less risk for human life due to the lack of passengers, as a result, automation on freighter's cockpit is going faster and further than in passenger's jets and the current visions on introduction of Unmanned Aircrafts in ATM system could arise sooner for the full automation of freighter than for the acceptance of the new military originated UAV's in the ATM system.

With a proper air-ground data link (proper means secure, reliable, high band...) on board automated systems could collaborate with on ground systems to guarantee a conflict free safe flight. With a proper air-air data link, even aircraft Flight Management Systems could collaborate to avoid collisions and to create common awareness of the shared airspace area, allowing total delegation of current pilot tasks to the automated systems. With enough bandwidth passengers could activate personal devices to work as if they were in the office or enjoy multimedia experiences and/or communicate within the aircraft and with mates flying in other aircrafts.

4.3.2 Road Transportation

Visions in this category should address the safety of road users and pedestrians. Often the envisaged sensor systems would gather data for real-time or close to real-time information services provided by governmental agency and private organizations including insurance companies.

Unlikely the ad-hoc and infrastructure-less characteristics, some of the visions require pre-established infrastructure of sensor nodes deployed in major roads. For instance, base stations every 1 to 5 Km and high-bandwidth backbone network. The sensor systems required vary from one scenario to another but it should include vehicle passing detector, structural material integrity, motion sensors and video capturing systems. Actuators are also discussed in a form of vibration, audible and visual (e.g.s LEDs). Most of such an information should be provided in real-time.

The **Sentient Guardian Angel** proposes the use of Wireless Sensor Networks to address dangerous traffic situations for elderly pedestrians, children as well as for disabled persons. Communication between the networks of the participants is used to detect the threat at an early stage giving adequate warnings using suitable audible/visual actuators, alerts and instructions to the ones involved.

With the goal of improving the road traffic, the **Supportive Road** vision describes scenarios where sensors installed on the roads assist in various traffic applications including road congestion avoidance and safety of drivers. It requires significant investment in technology to be installed on roads, which might only be available in the long-term. Similarly, the long-term vision **Congestion-Free Road Traffic** takes a step further to propose a technical solution to address traffic congestion. It explores the concept of dynamic time-space corridor that can be negotiated between cooperating vehicles to guarantee congestion-free journeys from departure to arrival.

4.4 Environmental Monitoring

4.4.1 Driving Forces

Below we discuss some driving forces that – as we believe – will promote the use of sensor networks for environmental sensor networks in the future.

Instrument for Earth Sciences: Earth sciences are interested in observing a large variety of environmental phenomena in order to understand, model, and predict them. Sensor networks are a new instrument that enables observation of phenomena that could not be observed before due to their remoteness, hazardousness, or sensitivity to instrumentation. Further, sensor networks enable an unprecedented spatial and temporal resolution of measurements.

Environmental Pollution: With increasing urbanization and industrialization, the environment and its inhabitants (flora, fauna, humans) are exposed to an increasing amount of pollution. Hence, it is important to monitor pollution, identify its sources, and enforce constraints on the disposal of pollutants. Sensor networks are seen as a valuable enabling technology in this context. In particular, there is a tendency that politics picks up this topic, issuing recommendations or mandatory regulations that prescribe monitoring of pollution levels or constrain the disposal of pollutants, thus pushing sensor networks as a technology to implement these regulations.

One example is noise pollution, which is not only unpleasant, but can raise the risk of diseases, negatively influences productivity and social behavior, and reduces the value of real estate. In the European Union, an estimated 300 Million citizens are exposed to noise pollution. Hence, the European Commission made the avoidance, prevention, and reduction of environmental noise a prime issue in European policy. For example, the European Directive 2002/49/EC requires member states to regularly provide accurate mappings of noise levels in urban areas. Sensor networks are seen as a prime technology to implement this directive [398].

Other types of pollution such as air, soil, and water pollution have been taken up by politics in a similar or even more aggressive way. Extrapolating this trend into the future, one can expect that even more types of pollution will be regulated and the strength of the regulations will further increase, thus resulting in an increasing need for sensor network technology to implement these regulations.

Natural Hazards and Climate Change: There is a huge demand for predicting the occurrence of natural hazards [265]. It is likely that climate changes will even increase this demand in the future. For example, it has been argued that glacial recession may cause dangerous rock slides [1] in alpine regions. Likewise, climate changes lead to decay of permafrost in alpine regions, potentially leading to increased impact and higher probability of rock falls [175].

In many cases, even very short-term predictions of such and other natural hazards require very detailed models of the underlying processes. Such models can only be constructed if the processes are monitored with a very high spatial resolution [265]. Sensor networks are seen as a key technology to implement such high resolution monitoring.

4.4.2 State of the Art

[172] survey a large number of environmental monitoring applications as reproduced in Figs. 4.10 - 4.12. They classify these deployments into four categories. *Large-scale single function networks* that tend to cover large geographical areas and take measurements for a single purpose, typically using large, expensive, wired, and line-powered nodes (e.g., weather stations). *Localized multifunction sensor networks* typically comprise smaller nodes and use more intelligent ad hoc networking. These systems typically measure simple scalar properties such as humidity or temperature that are useful for many applications. *Biosensor networks* networks are an emerging set of systems that use biological sensing elements, for example to measure the concentration of certain substances in air, water, or soil. Finally,

[1] http://www.spiegel.de/wissenschaft/natur/0,1518,427095,00.html

Example	Type of ESN	Sensors	Scale
Global Seismographic Networkhttp://www.iris.edu	Large Scale Single Function Network — seismology	Seismometer accelerometer	Global
The Georgia Automated Environmental Monitoring Network http://www.georgiaweather.net/	Large Scale Single Function Network — weather	Meteorological data	Regional
Web based hot spot modelling using GEOS http://goes.higp.hawaii.edu/	Large Scale Single Function Network — remote sensing	Multispectral imaging	Pacific rim and USA
Deep-ocean Assessment and Reporting of Tsunamis (DART) http://www.ndbc.noaa.gov/dart.shtml	Large Scale Single Function Network — tsunamis monitoring	Oceanographic and meteorological data+bottom pressure recorders	
SNOTEL http://www.wcc.nrcs.usda.gov/snotel/snotel-info.html	Large Scale Single Function Network — snow depth sensing	Snow thickness	Regional
National Science Foundation Polar UV Monitoring Network http://www.biospherical.com/NSF/default.asp	Large Scale Single Function Network — UV monitoring	UV	Polar regions
The Pacific Northwest Seismograph Network http://www.pnsn.org/welcome.html	Large Scale Single Function Network — Seismology	Seismic sensors	Regional
5 seismic projects http://www.cens.ucla.edu	Large Scale Single Function Network — seismology	Seismology	100 m–regional
Tropical Atmosphere Ocean Project http://www.pmel.noaa.gov/tao/index.shtml	Large Scale Single Function Network — oceanography	Oceanographic and meteorological data	
King County Lake Data http://dnr.metrokc.gov/wlr/waterres/lakedata/index.htm	Localised Multifunction Network — water quality	Weather, pH, conductivity, dissolved oxygen, chlorophyll	Local
Onondaga Lake Improvement Programme http://waterontheweb.org/data/onondaga/	Localised Multifunction Network — water quality	Temperature, dissolved oxygen concentration, salinity	Local
Olentangy River Wetland Research Park http://swamp.ag.ohio-state.edu/	Localised Multifunction Network — water quality	Weather, hydrodynamic sensors, webcam	Local
Ipswich-Parker Suburban WATershed Channel http://www.ipswatch.sr.unh.edu/index.html	Localised Multifunction Network — water quality	River flow, quality, precipitation, estuarine depth and quality, weather	Regional
8+ habitat sensing projects http://www.cens.ucla.edu	Localised Multifunction Network — habitat monitoring	Microclimate, video camera, with soil respiration (CO_2), nutrient flux (N, P, etc.)	100 m–1 km
Great Duck Island http://www.greatduckisland.net/	Localised Multifunction Network — habitat monitoring	Temperature, light, humidity	>100 m
Huntington Gardens http://www.sensorwaresystems.com	Localised Multifunction Network — habitat monitoring	Light levels, air temperature and humidity, soil temperature and soil moisture	1 km
Tucson Flooding Project http://www.sensorwaresystems.com	Localised Multifunction Network — habitat monitoring	Ambient air temperature, relative humidity, and light level. soil moisture	1 km
Sevilleta, New Mexico http://www.sensorwaresystems.com	Localised Multifunction Network — habitat monitoring	Light levels, air temperature and humidity, as well as soil temperature and moisture	1 km
Antarctica (analogue for Mars) http://www.sensorwaresystems.com	Localised Multifunction Network — habitat monitoring	Two soil temperature sensors in addition to air temperature, humidity, and light sensors	2 km
Lancaster Farms http://www.sensorwaresystems.com	Localised Multifunction Network — habitat monitoring	Light levels, air temperature and humidity, soil temperature and soil moisture	2 km
Malheur Experiment Station http://www.sensorwaresystems.com	Localised Multifunction Network — habitat monitoring	Light levels, air temperature and humidity, soil temperature and soil moisture	100 m
Sierra Nevada Mountains, California http://www.sensorwaresystems.com	Localised Multifunction Network — snowfall	Snow accumulation and melting	
Kennedy Space Center Launch Pad http://www.sensorwaresystems.com	Localised Multifunction Network — habitat monitoring	Light levels, air temperature and humidity, soil temperature and soil moisture	100 m

Figure 4.10: *Environmental sensor networks part 1 [172]*

Example	Type of ESN	Sensors	Scale
Cal Poly Pomona, College of Agriculture http://www.sensorwaresystems.com	Localised Multifunction Network — habitat monitoring	Light levels, air temperature and humidity, soil temperature and soil moisture	100 m
XYZ On A Chip http://www.cbe.berkeley.edu/research/briefs-wirelessxyz.htm	Localised Multifunction Network — HVAC monitoring	Airflow measurement and indoor temperature	Building scale
WAVIS (wave-current informationsystem) http://www.wavcis.lsu.edu/aboutus.asp	Localised Multifunction Network — Oceanography	Wave height, period, direction of propagation, water level, surge, near surface current speed and direction and meteorological conditions, webcam	Regional
MySound http://www.mysound.uconn.edu/index.html	Localised Multifunction Network — oceanography	Water quality, weather, wave data, webcam	Regional
Chesapeake Bay Observatory System http://www.cbos.org	Localised Multifunction Network — oceanography	Weather, salinity, wave speed, wave direction and conductivity	Regional
SECOAS http://envisense.org/secoas.htm	Localised Multifunction Network — ocean conditions	Location, wave heights	1 km
Argus (15 installed worldwide) http://www.planetargus.com/	Localised Multifunction Network — coastal erosion	Video camera	5 cameras 1 km
Floodnet http://envisense.org/floodnet/floodnet.htm	Localised Multifunction Network — flood warning	Water depth	1 km
CORIE http://www.ccalmr.ogi.edu/CORIE/	Localised Multifunction Network — fluvial observations and flood warning	Water temperature, conductivity, pressure, velocity, acoustic backscatter, wind speed and direction, air temperature and relative humidity, longwave and shortwave radiation	20 km
NWIS web water data http://water.usgs.gov/	Localised Multifunction Network — flood warning	Surface water, ground water and water quality	Regional
Volcán Tungurahua Project http://www.eecs.harvard.edu/~werner/projects/volcano/	Localised Multifunction Network — volcanic processes	Wireless infrasonic sensor array	>10 km
GlacsWeb www.glacsweb.org	Localised Multifunction Network — glacial processes	Weather, location, tilt, pressure, temperature	>100 m
Smart Gas-MIR space station air quality Persaud et al., 1999	Biosensor Network — air quality	20 element conducting polymer array	>100 m
Ferrera Air Pollution Monitoring Carotta et al., 2001	Biosensor Network — air quality	CO, NO, NO_2, O_3	>100 m
Cranfield University Sewage Works Bourgeois et al., 2003	Biosensor Network — water contaminants	12 conducting polymer sensors	>100 m
4 contaminant transport monitoring projects http://www.cens.ucla.edu	Biosensor Network — soil and water contaminants	Soil moisture and soil CO_2 sensors, nitrate flux	100 m
2 marine microorganisms projects http://www.cens.ucla.edu	Biosensor Network — monitoring algae	Immuno-based flow cytometry	Autonomous mobile robots
AWACSS http://barolo.ipc.uni-tuebingen.de/awacss/	Biosensor Network — monitoring water contaminants	Estrogens and progestogens in sediment and water	>100 m
UK Environmental Change Network http://www.ecn.ac.uk/	Heterogeneous Network	Weather, land and lake surface water discharge, camera	Regional
National Environmental Monitoring Initiative http://www.epa.gov/cludygxb/html/choices.htm	Heterogeneous Network — inventory of monitoring sites	Wide range	Regional
National Ecological Observatory Network http://www.neoninc.org/	Heterogeneous Network	Ecological monitoring	National
The Global Earth Observation System GEOSS http://www.noaa.gov/eos.html	Heterogeneous Network	Wide range	Global
SensorNet http://www.sensornet.gov/	Heterogeneous Network — Incident management system	Wide range	National
Coastal Ocean Observation Laboratory http://marine.rutgers.edu/cool/LEO/LEO15.html	Heterogeneous Network	Satellites, aircrafts, ships, fixed/relocatable moorings, and autonomous underwater vehicles used to measure a wide range of oceanographic properties	Local

Figure 4.11: *Environmental sensor networks part 2 [172]*

Example	Type of ESN	Sensors	Scale
NEPTUNE Project http://www.neptune.washington.edu/	Heterogeneous Network	Wired and wireless nodes+multipurpose robotic underwater vehicles to measure wide range of oceanographic properties	Regional
Orion Project, Scripps Institution of Oceanography http://orion.lookingtosea.ucsd.edu/	Heterogeneous Network	Oceanographic monitoring	Global

Figure 4.12: *Environmental sensor networks part 3 [172]*

heterogeneous sensor networks include the data sources from the other types of environmental sensor networks to monitor the environment at different scales.

4.4.3 Trends

Below we point out a number of technology trends that are becoming relevant for environmental sensor networks.

Use of Existing Infrastructures: As it is costly to deploy sensor networks, in particular at large let only global scale, it has been suggested to use existing infrastructure instead of deploying new networks. In particular, several recent projects investigate the use of mobile phones as networked sensors also for environmental monitoring [306], [464].

Reusable and Multi-Application Systems: Many environmental sensing applications use custom-built sensor networks that are tailored for one specific application. While different environmental sensing applications differ in sensing modalities and other aspects, they share many important characteristics. Hence, instead if deploying new hardware for every new application, it is preferable to have a reusable platform that can be parameterized for different applications and that can serve multiple applications at the same time. Examples for such developments are SensorScope [414], SNPK [433], or TASK [49].

Cyberinfrastructure: Existing environmental sensor networks typical use proprietary protocols, data formats, and data management systems to collect and store data sets. Hence, using these data sets is rather difficult as it requires exporting the data using proprietary tools, converting the data into a useful format, and importing into the user's data management system. In particular, this procedure complicates the correlation of multiple data sets provided by different parties. Moreover, the original data sets often miss important meta information such as the exact location and time where the data has been collected, systems and their parameters that have been used to generate the data, and so on.

To address these issues, a so-called CyberInfrastructure is needed to facilitate open and collaborative acquisition, management, and exploitation of environmental data sets. Recently, several efforts aim at providing elements of such a software infrastructure [448], [4], [220], [413]. Typically, such systems provide interfaces to a variety of data collection networks, convert the sensor data into a common format (including meta data) and store the data in a data management system, which provides primitives to access and process the data. Further, such systems provide advanced techniques for exploring, mining, and visualizing the data.

4.4.4 Challenges

[172] point out the following primary challenges for future environmental sensor networks.

- **Power Management**: Many environmental phenomena change slowly and require long-term or even permanent observation to obtain data that is useful to the scientist or user of the system. Hence, efficient power management schemes and energy harvesting schemes are needed to avoid frequent service of the sensor networks in remote regions.

- **Management and Usability**: Many systems are research prototypes, require substantial technical expertise to deploy and manage, and lack the robustness that is required for long-term operation in remote and hostile environments.

- **Standardization**: Today, interoperability of systems and data is hindered by a lack of standards at all levels, ranging from low-level communication protocols, over operation systems, to data and metadata representation.

- **Data Quality**: Better calibration is needed to obtain accurate data, the exact locations from where sensor data has been obtained is also often missing today, but needed in later analysis of the data.

- **Security**: As we deploy sensor networks for monitoring and predicting natural hazards, sensor networks need to be tamper-proof. Similarly, sensor networks to monitor the compliance with certain environmental regulations need to be secured to withstand possible attacks of those violating the regulations.

- **Data Mining and Harvesting**: An increasing number of long-term deployments, producing huge amounts of data available in a common format using common access methods will lead to a huge data set. To extract the interesting information from these data sets, advanced and easy to use techniques are needed to mine the data.

- **Optimized Sensors and Hardware**: New applications (such as monitoring environmental pollution) require novel types of biosensors, for example to measure the concentration of certain substances. Today, many of these sensors are quite voluminous and require substantial amounts of power (e.g., for heating). While miniaturization of sensor nodes may be an issue for some applications, future systems will likely require more powerful platforms to handle large amounts of data captured from advanced sensors (e.g., audio or video streams).

- **Cost**: Some sensors remain very expensive, for example, sensors for underwater use (e.g., pressure, temperature) currently cost more than 1000 €.

4.5 Healthcare and Assisted Living

The progress of science and medicine during the last years has contributed to significantly increase the average life expectancy. According to Eurostat, in 2050 life expectancy in Europe will be 79.7 and 85.1 years respectively for men and women [123]. The worldwide population over 65 is projected to increase from 500 million in 2006 to one billion in 2030. This trend is especially significant in Europe where the population over 65 is expected to become in 2030 24.3% of the population (compared to 12% worldwide) [465]. The increase of elderly population will have a large impact especially on the health care system. However, ICT technologies and in particular WSNs composed of wearable sensors can contribute to improve the quality of health care services by:

- enabling continuous monitoring of vital signs of patients even outside hospitals or care facilities
- supporting remote therapy and rehabilitation to reduce hospitalization costs
- allowing quantification of physical parameters for diagnosis and therapy monitoring
- providing doctors and caregivers with new data about their patients that was not previously available using traditional monitoring methods limited to clinical environment
- allowing to compensate increasing disabilities of an aging person through assisted living services
- providing records of personal health and fitness

WSNs support the provisioning of advanced remote and continuous monitoring services by allowing connectivity of sensor nodes placed on the human body and in the surrounding environment. Wearable sensor nodes allow to measure parameters such as heart rate, respiration rate, muscular tension, limbs acceleration. Instead, sensor nodes placed in the environment allow for example to detect smoke or occupancy. Remote connectivity is typically supported by a gateway node that interfaces the WSN with wide area networks.

Several health care and assisted living systems based on WSNs have already been implemented and in some cases deployed. In the next sections, we give an overview of the most relevant state of the art projects in the following application domains:

- assisted living
- activity recognition
- gait analysis
- emotion recognition

Then, we present design frameworks that have been proposed for healthcare and assisted living applications, and discuss trends and open challenges.

4.5.1 Assisted Living

Assisted Living systems can provide a variety of services to support humans in their everyday life, such as monitoring and control of appliances, detection of intrusions, emergency notification. Several projects such as [309] [496] [20] [193] have developed service architectures for assisted living services in the home. SK Telecom [99] offers the "Digital Home Service" that is based on wireless devices, sensors and services to monitor and control appliances, lights, smoke detectors, intrusion detectors, climate controls, gas valves and electronic door locks, all linked to the Internet wirelessly via a ZigBee-based residential gateway. Customers can monitor and control these devices remotely via Internet or mobile phone. Some assisted living projects specifically target elderly people. France Telecom and Ember have made trials of a system [115] that allows to monitor activities and support safety of elderly by providing a wireless "panic call" capability for emergency situations. The system also allows caregivers to monitor the state of exit doors remotely and identify unauthorized entries or departures. Fall detection systems [114] [461] based on accelerometers detect if an elderly has fallen and allow immediate call for rescue. Trilcenter [458] studies fall prevention techniques based on early detection of postural and neuro-cardiovascular instability. The objective is to enable prediction and prevention of falls through the measurement of neuro-physiological, behavioral and cardiac responses in the real-world environment.

4.5.2 Activity Recognition Systems

The recognition of physical activities is an important component of several health care and assisted living services. Wearable sensors such as accelerometers and gyroscopes allow to recognize body movements and activities with good accuracy [76] [337]. Several algorithms have been proposed to address this problem. A class of algorithms infers activities directly from accelerometer and gyroscope data. Accelerometers measure both the static vertical component, gravity, and the dynamic acceleration caused by movement whereas the gyroscopes measure the orientation based on the principles of angular momentum. Some algorithms find the orientation of the specific part of the body based on coordinate transformation to determine the angles of each accelerometer axis measurement with respect to the gravity when that part is static [352]. Other algorithms are based on calculating the distance traveled by certain parts of the body through double integration of accelerometer signals from which the gravitation component is subtracted. The gravitation component can be estimated by other sensors such as gyroscope [314]. The other way to estimate the gravitation component is using approximation algorithms based on such assumptions as not changing the orientation of accelerometer during movement or estimating the rate of change in accelerometer orientation [391]. Another class of algorithms is based on feature extraction and classification. Features are extracted from the observed sensor signals. Features may be simple statistics of the signal such as minimum, mean and energy or may be more complex functions. Techniques like principal component analysis or linear discriminant analysis can be used to select the most relevant features [103]. For activity classification several techniques have been proposed, such as thresholding [202], k-nearest neighbor [300] [401], decision trees [312] [324], neural networks [293], hidden Markov models (HMM) [219] [158], sparsity of vector solutions [504].

4.5.3 Gait Analysis

Gait analysis is used to study the locomotion of people and identify posture-related or movement-related problems. Gait analysis usually concerns the measurement of the movement of the body and the forces involved in producing these movements. [391] estimates the walking speed, incline and stride length by using a bi-axial accelerometer and one rate gyroscope. The gyroscope first measures angular velocity around its sensitive axis and allows to reconstruct pitch angle by integration where the initial angle value is estimated from accelerometer at zero velocity. The pitch angle is then used to remove gravitational component from accelerometer readings. Double integration of acceleration then gives the total displacement, which is used to calculate the stride length, walking speed and incline. [158] uses a HMM model to segment the gait into four different phases: stance, heel-off, swing, heel-strike. A genetic algorithm is then used to tune the model and decrease the error.

4.5.4 Emotion Recognition Systems

The problem of recognizing emotions has been addressed mostly by the HCI (Human-Computer Interaction) community [44] using facial [125] and voice analysis [50] techniques. However, the capability of Body Sensor Networks (BSN) to derive physiological parameters has recently enabled several research projects, which use physiological parameters measured by wearable sensors, such as ECG, accelerometers, EMG, body impedance, and skin conductivity to determine mental conditions such as emotions, mood, depression, attention level, stress and anxiety [274]. Skin conductivity is a physiological parameter that has been used in several research projects to measure arousal. In particular, the Affective Computing group at MIT Media Lab has developed a glove-like wearable device, called Galvactivator [135] that senses the wearer's skin conductivity and maps its values to a bright LED display. An increase of skin conductivity across the palm tends to be a good indicator of physiological arousal. Tokyo University [479] has developed a system based on galvanic skin response to recognize emotions and communicate them during online chats. Other projects have used sensors that capture parameters such as heart rate variability, blood volume pulse, breathing rate and volume. [492] provides a list and a description of parameters that can be recorded by a garment with embedded inductive plethysmography, called LifeShirt. [361] describes experiments conducted measuring ECG and respiratory activity of a group of healthy volunteers. These experiments provide preliminary evidence that basic emotions such as fear, anger, sadness and happiness are associated with distinctive patterns of cardio-respiratory activity. [210] [163] [449] present prototypes based on physiological sensors and their applications. IMEC's Holst Center has developed a BSN composed of two small wireless sensor nodes. The first is integrated in a chest belt and measures respiration and ECG. The second is integrated in a wristband and consists of a commercial sensor for skin temperature and a dedicated circuit board measuring the galvanic skin conductance between two fingers. The physiological measurements are combined and interpreted in the software running on the base station where an indication about the person's arousal is derived in real time [27]. The Aubade Project [15] has developed a wearable platform to monitor and recognize the emotional status of a person in real time using facial sensors. [179] describes experiments that use facial electromyography (EMG) as a measure of positive and negative emotional valence during an interactive gaming experience. [483] presents the physiological responses to different web page designs obtained by monitoring Skin Conductivity (SC), blood volume and heart rate (HR) of participants in various loosely controlled computer-based situations. The experiments in [286] measured multiple components of emotions in interactive contexts. To induce different emotional states, two versions of an interactive system were used which differed with respect to quality of use. The results suggest that systems of high usability lead to more positive emotions than systems with usability flaws. In [472] the authors present a user model associating psychological and physiological representation of emotion. For each subject, they estimated the position in the valence arousal space. The results showed that combining different emotion representations (dimensional and discrete, dynamic and static) into one User Model is suitable.

4.5.5 Design Frameworks

The design of BSN applications is complex and time-consuming due to the lack of proper abstractions that support interoperability and hide low-level details to application developers. Most applications require advanced algorithms to interpret the sensor data and derive patterns of behavior or health conditions. Classification and pattern recognition techniques [103] have been developed and applied mostly to other application domains. However, implementing them on BSNs has new challenges because BSNs are very resource-limited in terms of battery power, memory, and computing power. To meet the application requirements, designers must carefully evaluate implementation trade-offs regarding the allocation of resources, therefore they need flexible design frameworks that provide proper abstractions and support fast prototyping.

CodeBlue [289] is a framework built on TinyOS designed to provide routing, naming, discovery, and security for wireless medical sensors, PDAs, PCs, and other devices that may be used to monitor and treat patients in a range of medical settings. CodeBlue has been used to design several applications such as monitoring the limb movements and muscle activity of stroke patients during rehabilitation. The same group from Harvard University has recently developed a new operating system for sensor networks called Pixie [281] that enables resource-aware programming and allows applications to receive feedback on resource usage, and have explicit control over resources. Pixie is designed to support the needs of data-intensive applications, which involve high data rates and extensive in-network processing. The Pixie OS is based on a dataflow programming model and is based on the concept of resource tickets, a core abstraction for representing resource availability and reservations. By giving the system visibility and fine-grained control over resource management, a broad range of policies can be implemented. To shield application programmers from the burden of managing these details, Pixie provides a suite of resource brokers, which mediate between low-level physical resources and higher-level application demands.

Titan [278][278] is a framework, also built on top of TinyOS, that is specifically designed to perform context recognition in dynamic sensor networks. Context recognition algorithms are represented by interconnected data processing tasks forming a task network. Titan adapts to different context recognition algorithms by dynamically reconfiguring individual sensor nodes to update the network wide algorithm execution.

SPINE [155] is a framework that supports the development of signal processing intensive applications. It provides libraries of protocols, utilities and processing functions, and a lightweight Java API that can be used by local and remote applications to manage the sensor nodes or issue service requests. By providing abstractions and libraries, that are commonly used when signal processing algorithms are implemented in WSNs, SPINE also provides flexibility in the allocation of tasks among the WSN nodes and allows the fast exploration of implementation trade-offs. SPINE libraries and protocols support distributed implementations of classification algorithms that reduce the amount of data to be transmitted and save energy.

4.5.6 Trends and Open Challenges

Several applications enabled by wirelessly connected medical and fitness devices are already in the market [38] [461] [356]. However, to fully exploit the potential of WSNs in this domain and enable a broader range of advanced assisted living and health care services, additional research is needed to improve WSN applications with respect to parameters such as:

- Low energy consumption to maximize battery lifetime
- Wearability to allow patients carry sensor nodes in their daily life
- Privacy and security to ensure that only authorized people, e.g. doctors, relatives and caregivers, can access information on personal health or activities
- Low latency especially in life emergency scenarios
- Reliability, especially for applications monitoring vital parameters
- Accuracy in pattern recognition
- Easiness to extend the platform to other sensors and services, e.g. when new health care needs arise
- Provisioning of service across locations to support continuous monitoring.

In particular, future research in this domain will focus on two directions:

- Advanced pattern recognition algorithms for application scenarios requiring a more accurate identification of body movements or the fusion of multiple sensor data. The latest application scenario show a trend towards recognition systems that derive high-level parameters from multiple sensors data as well as context information. Defining robust and reliable multi-sensor data fusion techniques is one of the main challenges for the research community.
- Distributed implementation of signal processing algorithms on resource limited BSNs. This will become even more critical as the complexity of the algorithms and the amount of data they use increases together with the demand for greater miniaturization and longer battery life. Therefore, optimization techniques for efficient resource management and task allocation over the network will be needed together with flexible design frameworks and methodologies for fast prototyping and design space exploration.

Standards will play an important role in the definition of the market. IEEE 11073 [205] , ZigBee [531] and Bluetooth [36] are defining health care application profiles, while the Continua Alliance [80] will define guidelines based on selected standard interfaces for the interoperability among components of health care systems including sensors, application hosting devices and electronic medical records.

From a business perspective several directions can be followed:

- Research on a future care integration platform that combines smart devices and business services into business processes.
- Research on flexible and user centric device-service-composition based on a service repository (B2C) addressing the complexity of diverse service offerings.
- Research on ways to enable health providing organizations to easily expose, integrate or consume services (B2B) to establish broad health chains.
- Research on service brokerage and meditation and respective business models that satisfy regulatory requirements and ensure appropriate service quality (including SLAs etc.)

4.6 Security

Distributed cooperative objects, as explained and discussed in other areas, have great potential for future capabilities not presently achievable with existing technologies and lifecycle cost constraints. Cooperative objects form a solid technological ground for the development of easily deployable nodes that are capable of providing quality surveillance intelligence that can easily be translated into action.

An efficient surveillance network should effectively provide the following capabilities complying with predefined performance parameters and at a reasonable cost:

- Detection of targets
- Identification and classification of targets
- Tracking of targets
- Communication with higher level systems
- Situational awareness for decision makers

By using a network of cooperative distributed objects, many of these functions can be leveraged and improved in many aspects. The following discussion presents a preliminary vision of where and how these improvements could take place. Note that we are not addressing here the potential for new applications for networked Cooperating Objects, but how existing applications would benefit from such concept. New applications will certainly be possible, although this will necessarily come once the concept has been materialized and field proven in already existing applications.

4.6.1 Deployment

One of the aspects that make cooperative objects unique when confronted to their self contained counterparts is the size of the individual elements to be deployed. An array of basic performance multi-sensors may achieve the same performance a single state-of-the-art unit; it is just a matter of how many of these reduced units have to be deployed. Because cooperative objects will typically integrate a reduced number of functions in the same structure, their size is likely to be smaller. This brings two key advantages when it comes to deployment:

1. the individual unit's footprint is smaller, thus easier to place and disguise

2. deployment time is accordingly shorter

Obviously these advantages are partly countered by the fact that more individual units need to be deployed for an equivalent system's performance.

4.6.2 Data Acquisition

Reduction of the size of units also allows for the possibility of deployment closer to the source of the events to be sensed. If for instance a wide area of cooperating IR or radar sensors are deployed in place of a single IR or radar unit, it is very likely that the average distance between the target and the closest sensing unit in the network is now smaller than the average distance between the target and the single unit case. By being closer to the source, the impact of clutter will be reduced since now the sensor has a stronger signal to discriminate against unwanted background noise, interfering signals and neutral targets.

Another aspect that we envision as a unique property that only distributed sensors can provide is stand-by environment sensing. During the time a threat is not present, the sensors can learn from the environment in order to be better prepared to detect and correlate abnormal situations in the future. These stand-by sensing includes event sequences and cycles (traffic of vehicles and people, power surges...), environment properties (daylight cycles, ambient temperature, background acoustic noise, EM activity, vibration...) and possibly many others as well. This very local knowledge will allow smart sensors to better discriminate the presence or occurrence of a threat in the presence of the very specific and unique clutter within the local environment of the sensing unit.

4.6.3 Distributed Data Processing

A property that is not present in self-contained units, is the possibility of performing distributed data processing. In a distributed network, each sensor will generate raw data products, or perhaps slightly processed in order to reduce bandwidth usage; however, these data aggregated from all sensors need not be processed by a single higher level unit, that is, there may be higher level units cooperating in the function of data processing. The benefits for such an approach can be materialized by applying different correlation algorithms or by implementing a Kohonen neural network system.

Additionally, higher level objects can be included to further process the data and present it at an application or session level, getting closer to a situational awareness presentation that is more human-like and less dependent on sensor specifics.

4.6.4 Lifecycle Cost

Perhaps one of the most often used arguments in favor of distributed Cooperating Objectsis the potential for reduced costs in the light of current trends in miniaturization, reduced power consumption and economies of scale. We are not going to discuss at this point whether the purchasing price tag for future distributed cooperating sensors would be higher or smaller than that of a conventional system, however there some aspects of lifecycle costs that are worth looking at.

First, a network of distributed sensors is likely to need a less rigid maintenance schedule than its self contained counterpart; the reason is that the network will be less sensitive to sensor de-calibration of individual units, and thus maintenance can be arranged at more convenient times in order to keep such operating costs to a minimum.

Second, it is also very likely that individual sensors needing attention could easily be substituted by fresh ones, thus reducing field service time and cost. Additionally, the increased simplicity of each sensor will allow for a more efficient recalibration or repair service time since less personnel specialization would be required.

4.6.5 Performance

Performance of a network of distributed Cooperating Objects is something that we cannot yet assess due to the high uncertainty involved around future developments, however the future looks bright. Within the most relevant performance aspects to be addressed (other than sensor precision and resolution), we could highlight reliability, vulnerability, endurance and disposability.

Reliability of a network of Cooperating Objects can significantly be higher than that of a self contained system. The reason is that it allows for network architecture that presents far less (if any) single points of failure. A simple RAM & FMECA analysis will reveal such strength easily.

Vulnerability, following the same arguments as for reliability, will benefit from a networked architecture. The network will show increased robustness against attacks. Additionally, the reduction in the size of units will allow for more compact devices that will be more difficult to detect or neutralize.

Endurance is perhaps the weakest point with current technology. The miniaturization of electronics is evolving much higher than the miniaturization of its respective power supply requirements, particularly the size of any batteries or auxiliary power sources (solar panels, fuel cells, etc).

Finally, networked Cooperating Objects will be easier to dispose. Disposing may include recycling, destruction of sensitive data or safe treatment of toxic components; all of them will benefit from reduced size and simpler construction.

4.7 Conclusions

It seems clear from the wide range of applications presented in this chapter that the reach of Cooperating Objects technologies extends to almost every aspect of our daily lives. However, these are just the tip of the iceberg if Cooperating Objects manage to become one of the mainstream technologies in the next years. In general, all applications that can increase performance of functionality from the cooperation of smart embedded devices, will benefit from the line of research followed by the Cooperating Object community.

Nevertheless, in order for this to happen, the market has to be ripe for the technology and the appropriate research gaps will have to be closed to a level that can be used by the industry in order to create products and applications that can be installed in the real world. We will try to answer both of these questions in the next chapters.

Chapter 5

Market Trends Overview

The domain of Cooperating Objects is still at its dawn; however its impact is estimated to be broad and significant that could drastically change the future application and services. Numerous market analyses seem to “confirm” this trend. It is important to understand that Cooperating Objects is a huge domain, and as such it is very difficult to set the limits and estimate its total value. As such we indicatively refer only to some markets that fall in the category of the Cooperating Objects such as the (wireless) sensors, embedded systems etc.

5.1 Market Predictions

According to ON World Inc., the global market for Wireless Sensor Network (WSN) systems and services is expected to skyrocket to about $4.6 Bn in 2011, up from approximately $500 million in 2005. There will be a worldwide (conservative estimate) market of $5.3B for the industrial control segment only, comprising 4.1 Million nodes by 2010. ON World Inc. most aggressive forecast for all wireless sensor (& control) network segments is $8.2B by 2010, comprising 184 Million deployed nodes. It is important to note that ON World Inc. projections only account for the physical node hardware shipments - not the physical gateway hardware, nor any independent system software components, enterprise software components, system integration services or other ancillary services. Hence, their forecast for WSNs does not capture all areas of revenue.

A recent market report by Frost & Sullivan claims that even though wireless sensors are rapidly gaining ground in industrial sectors such as building automation and industrial automation, the adoption of this technology has been relatively slow. Educating and convincing end users about the various advantages of wireless sensors will be critical in increasing adoption levels. Further, the study mentions that the total wireless sensor revenues were $160 million in 2005 and it can reach $1850 million in 2012.

	Million €	**Market Structure**	**CAGR 2007-2011**
Industrial	2643	25.9%	7%
Automotive	2304	22.6%	3%
Aerospace/Defense	1684	16.5%	8%
Laboratories/Test	952	9.3%	4%
Consumer	724	7.1%	2%
Medical	534	5.2%	16%
Security	250	2.5%	5%
Transport	226	2.2%	5%
Building	204	2.0%	7%
Energy	199	2.0%	4%
Home Appliances	183	1.8%	2%
Environment	160	1.6%	15%
IT Infrastructure	138	1.4%	4%

Table 5.1: *European Sensor per Application Sector [122]*

The firm "Research and Markets" predicts that the market for wireless sensor systems should grow rapidly over the next 5-10 years . Depending on the outcome of standardization efforts and developments in affiliated markets, sales of wireless sensor systems could reach $5B to $7B dollars in future. The WSN market must grow rapidly to reach such levels so quickly, however. As the market takes off from its current small base, sales are predicted to multiply year-to-year settling in at an annual growth rates at a still substantial 40 or 50%.

In their 2006 research on the Extended Internet, Forrester Research predicts revenues for Wireless Sensor and RFID Networks hardware/software/services to be over $3.7 B in the industrial automation and maintenance and physical control and security segments. Their estimate of current revenues in this space for 2006 is only $138 M. Hence, they are predicting rapid growth.

In 2012, the Wireless Sensor Network (WSN) smart home market will be worth $2.8 Bn worldwide, up from $470 million in 2007, estimates Electronics.ca Research Network . The "smart home" is becoming a reality for the mass market as hundreds of products currently shipping and established service providers such as AT&T and SK Telecom are starting to offer WSN based home monitoring services.

According to BCC Research (www.bccresearch.com), the global market for microsensors will increase from $2.7 Bn in 2007 to an estimated of $8.4 Bn by 2013. Furthermore biochips' market share is expected to increase from 17.2% in 2007 to 21.6% in 2013.

US demand for chemical sensors is projected to surpass $5 Bn by 2012. Biosensors will continue to be the largest type of chemical sensor, as the increasing number of diagnosed diabetics continues to boost demand for blood glucose test strips.

The world market for technologies, products, and applications alone that are related to the "Internet of Things" will increase significantly from 1.35 € Bn to more than 7.76 € Bn in 2012, with 50% average growth rate annualy [400].

5.2 Cooperating Objects for Monitoring and Control

The main focus of Cooperating Objects is in coupling the physical and virtual worlds; they do this via monitoring and control activities. The European Commission has conducted an extensive market report [122] with respect to the Monitoring and Control (M&C) area (Table 5.2).

Hardware	Control (PLCs, NCs, power switches ...), Interfaces (PCs, HMI, ... Network, Computing Systems, OS & drivers
Software	Communication software, application and visualization, Development, Simulation, modeling, Decision Support Systems, ERP ...
Services	Application design, integration, installation and training, communication and networking ...

Table 5.2: *Products & Solutions for all Monitoring & Control markets*

The most important application market sectors identified are depicted in Table 5.3. The worldwide market for Monitoring & Control products and solutions estimated to be approx 188 € Bn. With a share of 61.5 € Bn, Europe represents 32% of this market. Services, with more than 50% of the market value, have the biggest share. The size of the worldwide and European monitoring and control market is distributed as seen in Figure 5.1.

Together, three application markets, Vehicles, Manufacturing and Process industries represent 60% of total Monitoring & Control market. Also Healthcare, Critical infrastructures and Logistic & transport follow closely. At the moment Home is still considered a small niche market. Between 2007 and 2020 the European monitoring and control domain is expected to grow at a 5.7% rate annually.

5.3 Market Trends

The overall market where Cooperating Objects technologies are contributing is expected to grow significantly until 2020 (see Figure 5.2). Hardware is expected to have a relative small growth due to decreasing prices; this does not hold true for network devices which will have an exponential growth in the next years. Software and services will have a higher growth than the average total market mainly due to the high growth of :

Environment	Waste treatment, Mining, Oil & Gas, Forestry, Air & Pollution, Agriculture, Fishing, Aquaculture ...
Critical Infrastructures	Airports, Harbors, Roads, Rivers, Communication Networks, Water Supply, Pipelines, Heat & steam, Gas Transportation, Storage & Distribution
Manufacturing Industry	Manufacture of: Computers, Electronic and Optical Products, electrical equipment, machinery, motor vehicles, trailers, transport equipment ...
Process Industry	Food, Beverage & Tobacco, Textile, Leather & Wearing, Paper, Printing, Petrol & Cook refining, Chemistry and Pharmacy, Rubber & Plastic, Mineral products (Glass, cement, ... & other non-metallic mineral products), Metal, ...
Building	Construction, Renting & Operating of estate, Tertiary sector ...
Homes	HVAC, Alarms, Lighting, Access control, Motorization ...
Household appliances	TV and other audio video products, White goods (washing machine, fridge, etc) ...
Healthcare	Human health activities (hospitals and doctors material, ...), Residential care activities (retiring residents, residents for disable people, ...), Nomadic medical equipment, Personal medical equipment, etc ...
Vehicles	Rolling stocks, Ships, Aircrafts, Cars, Off-roads, Agriculture, etc ...
Logistics & transport	Mail/Courier services, Goods handling, Warehouse, Ticket & Traveling, Fleet management, etc.. . .
Power Grids	Power plant (nuclear, oil-fired, coil-fired, ...), Electricity transmission, Electricity distribution (metering included), Electricity retail, Microgeneration, etc. ...

Table 5.3: *Application Sectors*

- Communication and networking,
- Simulation and modeling,
- Decision support and ERP,
- Integration

A detailed analysis of the state of the art as well as trends in the Cooperating Objects that will contribute to these market growth factors can be found in Chapter 3 and Chapter 6 respectively. Services are expected to dominate the market i.e. next generation of products or components is included in service packages. This emergence of new services

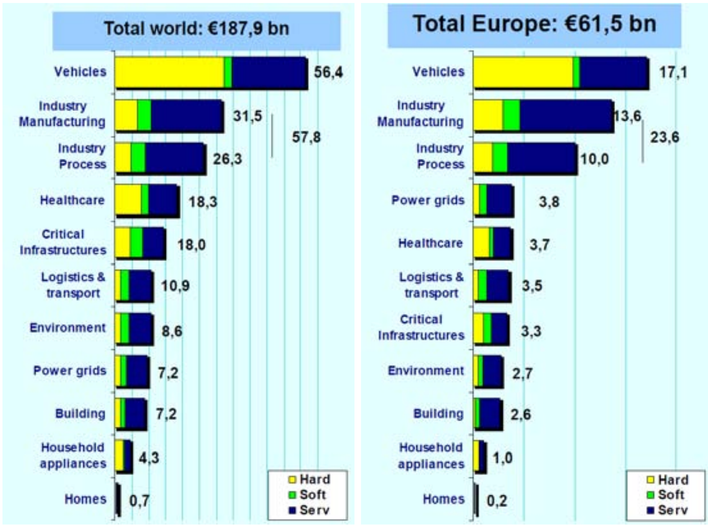

Figure 5.1: *Hardware Software and Services in European Monitoring and Control Market [122]*

will create also the need for next generation products e.g. in environmental regulations, Energy efficiency etc. The Total Cost of Ownership (TCO) is expected to be extended and include issues such as precision maintenance, asset management, production tools life extension with higher maintenance needs, more secure & safe installation & infrastructures. world M&C market is expected to grow from 188 € Bn in 2007, by 300 € Bn, reaching 500 € Bn in 2020. The M&C European market follows the same trends as the M&C world one in terms of product repartition and also markets products evolutions. The European M&C market will be reaching 143 € Bn in 2020.

Several Innovations are expected. In Components, increasing computing power and integration, intelligent communicating local components, standardization and lower prices are foreseen. In Networks, IP will be everywhere, networks will be transparent across application sectors, and service oriented approaches will be dominant. On the Services, it is expected that we will have a largely industrialized version of them.

As most of the technologies are already in place, what remains is the optimal exploitation of them. Many technologies still seem futuristic and with prohibitive cost for mass-application usage. As such the evolution of the domain wont be heavily based on the

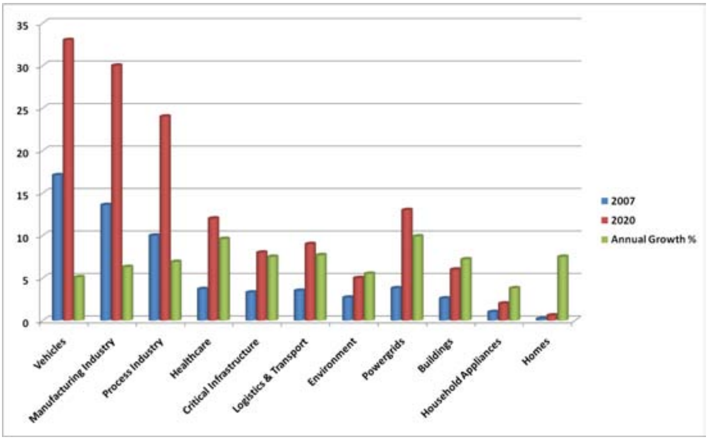

Figure 5.2: *Monitoring and Control Market 2007-2020 [122]*

technology as such only but directly linked to different business models which are connected to it.

5.4 The CONET Survey

The data presented in the following graphs were gathered via an survey done by the CONET Consortium. The survey was carried out online and at selected events. The distribution of the organizations that participated in the survey is depicted in Figure 5.3. As it can be seen there are representatives from all domains, with of course academia and research centers dominating the participant list due to the nature of the events that the survey was distributed. Nevertheless there is a concise and clear trend on the results. We will not focus on the details of each specific domain and the timeline for it, as this is analyzed in detail in Research roadmap (chapter 6).

Several domains are to acquire the thrust of benefits from Cooperating Objects. We have found out (depicted in Figure 5.4) that especially monitoring and management in Automation, Environment and Health followed closely by the energy, transportation and logistics domain will be the major beneficiaries . These findings are in line with the M&C market study presented in section 5.2, where again the automation industry is leading.

Several standards are under consideration for usage in Cooperating Objects. Some of

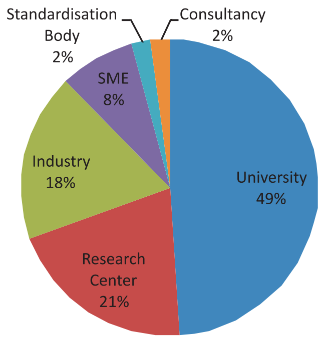

Figure 5.3: *Survey: Organization Distribution*

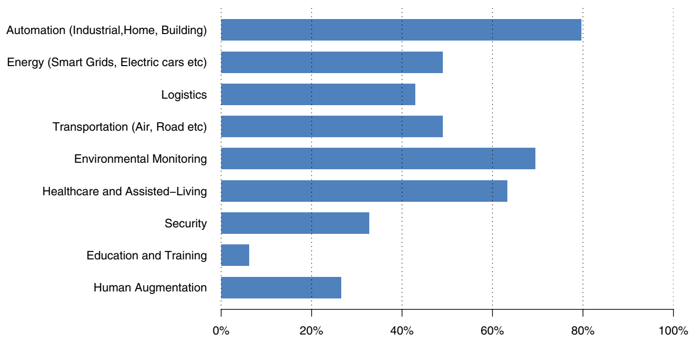

Figure 5.4: *Survey: Beneficiary Domains*

them are depicted in 5.5, where it is clear that pure communication standards 6LoWPAN, ZigBee, WirelessHART are the best known and are seen as important for the further development of Cooperating Objects area. However, the rest of the standards and efforts depicted such as DPWS, OPC-UA, REST etc are all over IP and deal with the interaction of Cooperating Objects at a higher layer. As we have seen many times, the second ones gain more importance and are in the focus of the community once the basic connectivity protocol issues (represented by the first group) are solved. Several other standards e.g. Bluetooth, WiFi, UWB etc might play a role here, and we plan to integrate them in future surveys.

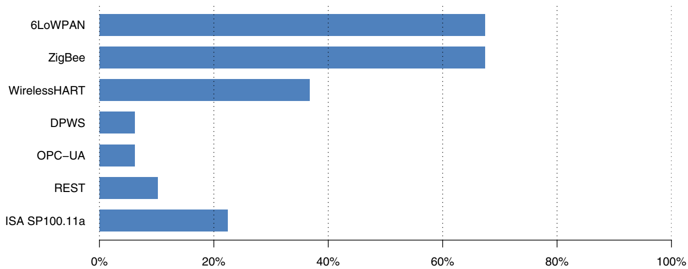

Figure 5.5: *Survey: Standards Impact*

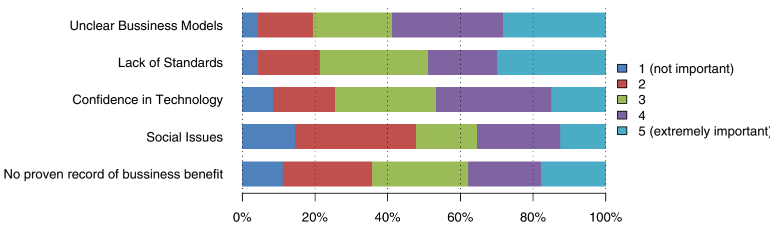

Figure 5.6: *Survey: Roadblock Impact*

For the wide-spread adoption of Cooperating Objects technologies in mass-market products, several roadblocks are identified (depicted in Figure 5.6. Unclear business models remains a critical issue to be solved, while lack of common standards (at all layers not only on communication) is still an issue. Confident in technology is seen as important but not as critical and should be addressed. Furthermore social issues and no proven record of business benefit are seen as having a moderate effect on the success of Cooperating Objects. Especially the last one is typical in the technology domain as the advances and benefits can not be fully envisioned nor widely understood.

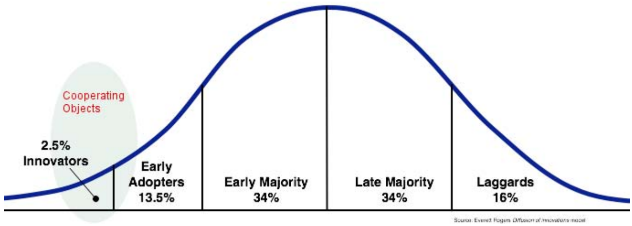

Figure 5.7: *Cooperating Objects in technology adoption lifecycle*

5.5 Conclusions

The majority of the market growth predictions were made before the economic meltdown of the late 2008 and 2009. As such, the aforementioned numbers should be taken as an indicative trend in the market and show its potential; the future will tell if and at what timeline they will be validated.

Nevertheless, it is clear that there is promising potential in versatile domains, that could greatly benefit with the introduction of Cooperating Object technologies, ranging from automation (home, industrial, building) to healthcare, energy etc. We estimate that we are stil in the dawn of a new era, and in the early phases of Rogers' technology adoption lifecycle as depicted in Figure 5.7. We expect that the Cooperating Objects market will be cross-domain and strongly embedded in the fabric of success of other domains.

Chapter 6

Research Roadmap

Using as a basis the analysis of the state of the art of Chapter 3, the review of innovative applications of Chapter 4 and the market analysis performed in the previous chapter, we present now the trends and gaps in Cooperating Objects research. These will then be used in Chapter 7 to identify the predominant areas that will need attention in the next years. Additionally, we discuss the results of our own estimation and the one from a series of surveys conducted among experts that indicate the approximate time where these gaps are expected to be solved.

6.1 Gaps and Trends

The data presented in the following graphs was gathered via an initial survey done by the CONET Consortium (see Section 5.4). For each of the research areas we asked the participants to rate the importance of this area from 1–not important to 5–extremely important and to give an estimation when they expect these issues to be solved.

In order to ease up the reading, we have followed the same structure as in chapter 3. For more information on a specific topic and not just on the trend or gaps that can be seen in it, please refer to the aforementioned chapter.

6.1.1 Hardware

The state of the art section as well as the CONET survey (see Figure 6.1) highlight that the current major gaps with regards to hardware for Cooperating Objects are power efficiency and energy harvesting. The following section considers these gaps in greater detail. We expect that as hardware is getting cheaper then the offer on new sensors with increased capabilities as well as low-cost devices will increase. Miniaturization is important in some areas e.g. healthcare, and is interesting, however the priority is given to fulfilling the main goals e.g. energy and cost-effectiveness.

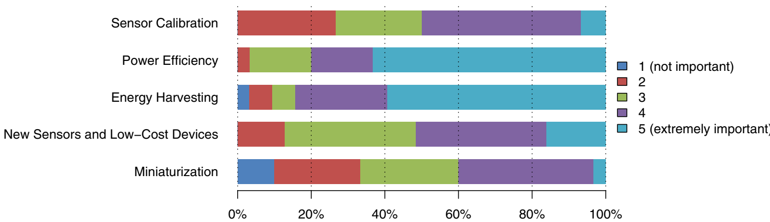

Figure 6.1: *Survey: Hardware*

6.1.1.1 Power Efficiency

The current trend we are seeing is the move toward smart devices that are more energy efficient. Where it is feasible increased operational longevity is achieved by using less capable mode efficient sensor nodes to cue other more capable but less efficient nodes that may be "sleeping" during periods of inactivity. As described in the state of the art section there is no one area that will provide the complete answer to improved longevity, but it is something that must be addressed at a system level where things like more efficient content/context aware communications, smarter algorithms and soft sensing (i.e. sensing things by measuring other things within an environment) all have a part to play.

A greater level of integration is required between the Motes major functional elements (ADC, microprocessor/controller, radio, DSP, storage, power management and energy harvesting) than is present in today. In terms of the platform itself we are looking for higher levels of integration on the processor and processors where we can switch off functions within the chip when they are not required. The other areas of chip level design that could help with system level efficiency are dynamic voltage and frequency scaling where processor clock and supply voltages could be dynamically adjusted to reduce node power.

In terms of power saving within processing devices Cooperating Objectscan look to solutions from the specialized medical implant domain. An example of this is the recent reporting of a novel technological approach (named "AMx" technology) developed at Imperial College that provides a potentially useful new signal-processing paradigm. The approach is based on the observation that most of the power consumption in conventional signal processing is concerned with data reduction. It is not unusual for mega-byte quantities of raw data to be processed in order to extract just a few bytes of meaningful information. Thus there are huge savings of power and gains of speed to be made if most of the data reduction can be made before digitization. The down side is of course the need for application specific signal processing hardware rather than general purpose digital signal processing. The Imperial College approach addresses this issue by developing a methodology based around analogue signal processing within custom silicon CMOS devices operating in weak inversion, at low voltages, on a thermally activated density of electrons in the channel to get 'logarithmic' amplifier performance. To exploit the very different operating characteristics of these devices novel signal processing has also had to be developed. Currently processors using this approach are not that capable but with time we could see this situation changing.

Architecturally it may also be possible to improve processing hardware efficiency, and here the area of analogue computing could be exploited (modern electronics may provide of us an integrated programmable solution and as far as a system level up-take is concerned it could be cost effective to incorporate analogue computing techniques within specialized chips that do the front end sensor interfacing for a WSN node). As well as the possible exploitation of techniques such as analogue computing at the sensor interface level it may be appropriate to look a distributed approach where we interface the sensor device with a less capable processor such as a PIC and only enable (awaken) the more capable node processor when the system has determined it has something useful to do.

6.1.1.2 Energy Harvesting

Another important hardware gap for Cooperating Objects is that of better (more capable and smaller) power sources. Power sources encompass the broad topics of battery technologies, fuel cells and energy harvesting. Developments in this area will come from new battery chemistries and from technologies such as the Edinburgh University Microbiobial fuel cell.

As stated in section 3.1.3 the energy harvesting technologies (e.g. thermal energy, electromagnetic energy, etc) each currently only suits certain application scenarios, and some have yet to produce useful amounts of energy for practical application. Current research in the field of energy harvesting tries to combine existing techniques to create more efficient power generators, although there is definitely the need to improve the energy generation capabilities of individual techniques. New materials, such as electroactive polymers, are being examined since they promise a higher energy conversion coefficient.

6.1.1.3 Radio Resource Management

A major research area concerns the design, analysis and implementation of resource allocation and adaptation strategies. One line of work could be the design of schemes which are provably optimal for synthetic interference models (see Section 6.1.5.1), another line of work the design of schemes which perform well under a wide range of channel models including the experimental ones. An important part of this research area includes the design and assessment of lightweight estimation schemes for the channel quality indicators (interference, packet loss, etc.). Another line of research in this area concerns the adoption of cognitive radio. Cognitive radio is a very interesting technique to circumvent external interference, but it requires relatively complex physical layers that cannot easily (and cheaply) be used on wireless sensor nodes. Therefore, there is a need for designing cheap and efficient cognitive radio schemes that can be used on sensor nodes.

6.1.2 Algorithms

In the algorithms section we consider the minimal set of functions required for the proper performance of Cooperating Object networks. Among these algorithms we include Querying, MAC, Data Storage and Localization.

Our survey reveals a general feeling that problems on the area of functional properties have been solved to a large extent, but there is still some work to be done. In Figure 6.2, the survey results reflect the importance of each area as perceived by the community. A high importance reveals that there are still significant challenges faced in that particular area, while a low importance reflects that most of the challenges have been tackled. Hence, the importance in our survey does not capture the importance of the role itself but rather the importance of exploring the problems in the area.

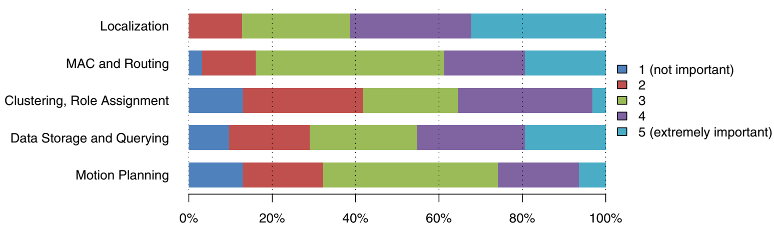

Figure 6.2: *Survey: Algorithms*

Multiple sinks, Motion planning and Clustering reflect a particularly low importance, which suggests that less research efforts should be directed to these areas. On the other hand, the survey shows that Data Storage, Querying, MAC and Routing still need some more research work, in spite of the significant body of literature present in these areas. Finally, Localization appears as the most important topic in the context of functional properties.

In the next sections, we explore in more detail the research gaps on these areas as well as other areas related to functional properties.

6.1.2.1 Localization

Localization has been, and still is, among the most active research areas in Cooperating Objects. Most of the localization work has been focused on two main directions: (a) allocation of physical coordinates for nodes or objects in the network and (b) target tracking. While notable contributions have been made in both of the directions mentioned above, there are still some open research problems to be completed.

Allocation of Physical Coordinates. Several works in the area of localization assume the existence of additional hardware such as microphones and speakers. These solutions provide accurate localization due to the ideal characteristics of sound waves but are expensive. When nodes do not have specialized hardware, radio signals are the most common tool to derive localization. Radio signal based localization has been extensively studied for static networks without interference. However, localization in mobile networks has not received the same degree of attention and remains an open area of research. Several applications of Cooperating Objects will involve mobile nodes willing to communicate with each other (such as inter-person communication in body sensor networks), in these scenarios, interference will lead to stochastic effects on the received signal strength, and hence, to significant localization errors.

A promising new technology for localization is Ultra Wide Band (UWB) technology for ranging. The ultra-wide frequency spectrum of UWB impulses permits a very fine time-resolution at the receiver, which can be exploited to estimate the distance between transmitter and receiver. Some first commercial products like Ubisense are available. However, the integration of UWB devices into small Cooperating Objects technology is still in a development phase.

Target Tracking. In many scenarios it is required to identify the presence and movement of an object. In target tracking, nodes not only need to be aware of their location but they also need to perform localized algorithms (with their neighbors) in order to estimate the position of the object of interest. While important contributions have been made with simple binary detectors, scenarios with (a) noisy measurements, (b) potentially high speed of object and (c) presence of multiple objects have not been fully explored. Given that one of the main applications of Cooperating Objects networks is expected to be surveillance, it is central to develop algorithms for the scenarios mentioned above.

6.1.2.2 MAC and Routing

The development of Medium Access Control (MAC) protocols is influenced by the capabilities of the radio transceiver, henceforth, MAC protocols have to be revisited with the appearance of each new generation of radios.

In recent years the community has moved from byte-level interfaces such as the CC1000 chip to packet-level radios such as the CC2420 chip. This switch brought advantages and new challenges. On one hand, packet-level radios reduce the load on the micro controller because it has to handle packets instead of individual bits, which leads to lower energy consumption. However, on the other hand, packet-level radios limit the use of low-power listening techniques because packet-level radios can no longer control the length of the preamble. Low power listening plays a central role in reducing energy consumption, and hence, it is important to develop efficient techniques for the current and next generation radios.

Another important characteristic of radio transceivers concerns the use high-speed radios with more sophisticated coding mechanisms. These radios too have advantages and disadvantages for low-power networks. On one hand, high speed radios provide better energy-per-bit ratios and sophisticated coding techniques permit lower signal-to-noise ratios, which implies less transmission energy and/or greater coverage. On the other hand, the complex receiver circuitry makes idle-listening a costly task. For example, the CC2420 radio consumes more energy when receiving than when sending (63 vs. 57mW). These changes ask for the re-design of several algorithms aimed at earlier generation of radios. Also, higher-data rates radios have a higher relative cost when switching the radio between send and receive mode which may impact the functional parameters of various MAC protocols in the literature.

In terms of architectural design, a recent trend is to combine the flexibility of random CSMA-style MACs with collision-free TDMA-based MACs. CSMA-based MACs are simple and efficient for low traffic scenarios but perform poorly under high contention, while TDMA are more complex mechanism for high traffic but it increases delay in low traffic scenarios. Individually, each of these mechanism have been studied thoroughly and no major breakthroughs are expected. However, the combination of these mechanism have achieve some interesting results and is expected to continue in the future. For instance, Z-MAC implements a TDMA overlay on top of B-MAC and switches dynamically depending on the level of contention.

Another important open area in MAC research is the accurate assessment of the myriad of protocols proposed in the literature. The performance of MAC protocols depends largely on the application, environment, as well as the specific hardware platform. Hence, it is important to filter the most ideal candidates, compare their mutual performance in various scenarios and provide a taxonomy that could help the end user in selecting the ideal MAC protocol for his/her application.

6.1.2.3 Available Bandwidth Estimation

Currently, networks of Cooperating Objects have limited radio capabilities. However, MEMS development suggests that bandwidth will increase and Cooperating Object networks will be able to deliver high-bandwidth delay-sensitive content such as video and audio. In these scenarios it will be central to estimate the end-to-end bandwidth in order to develop proper admission control policies.

The main challenge to be faced by bandwidth estimation techniques in Cooperating Object networks and especially WSN is the limited energy resources available on each node.

Most of the bandwidth estimation techniques on MANETs and the Internet use numerous packet probes. In Cooperating Object nodes will be allowed to use a very limited number of packet probes to estimate their bandwidth, which asks for light weight and possibly indirect mechanisms to estimate the available bandwidth.

Accurate bandwidth estimation in networks of Cooperating Objects is not only challenging due to the limited energy resources but also due to the dynamic conditions of the channel. On traditional networks, the available bandwidth is affected only by the amount of traffic. On wireless networks not only traffic but also the channel conditions affect the capacity of the link. Considering the variety of scenarios targeted by Cooperating Object networks and the simple radio transceivers used for communication, the problem of bandwidth estimation is significantly challenging.

It is important to remark that, up to now, there has not been much activity on the problem of bandwidth estimation in Cooperating Object networks.

6.1.2.4 Clustering

Node clustering has been a widely discussed research topic in the community of wireless sensor networks, and has been utilized by many algorithms in various aspects, specially in conserving energy consumption to prolong system lifetime. In addition, clustering offers a virtual hierarchical infrastructure for collaborating nodes to achieve scalability.The main research topics of node clustering include: (1) finding a minimum set of collaborating cluster heads while the connectivity among cluster heads is still maintained; (2) designing scheduling algorithms for intra- and inter-cluster data transmission, and (3) maintaining the connectivity among cluster heads by either adjusting transmission power or rotating cluster heads.

Because of the popularity of node clustering, challenging issues related to above research topics have been mostly explored and tackled in the literature. While most research efforts focus on static or dynamic cluster head selection and connectivity maintenance, less attention has been paid on characteristics of clusters in order to minimize the interference caused by topology structures.

For instance, clusters with equal size help in balancing the work load of cluster heads; *solid disc* property of clusters addresses that each cluster member is located within a constant distance to its cluster head, and thus reduces interference with neighbor clusters – assuming all nodes transmitting with the same maximum transmission power. Exploring these characteristics of clusters can greatly improve cluster construction in order to achieve higher energy efficiency.

6.1.2.5 Querying

Querying has been perhaps the most active research area in Cooperating Objects, especially in the domain of wireless sensor networks. While we do not expect any major paradigm breakthrough, there is still a need to filter and fine-tune the practically-useful algorithms from the unrealistic ones. In particular, the next stage of querying research in Cooperating Objects should be focused on testing scalability (networks in order of one thousand nodes and above) and the performance in real deployments.

Scalability has been the quintessential claim of Cooperating Objects networks, however, minimal experiments have been made in networks consisting of 1000 nodes or above. In order to quantify the realistic performance of next generation networks we need to test the reliability and efficiency of querying algorithms in real large scale networks. Several works have argued that flat (same node-type) networks will not scale and multiple-tier networks will be needed. However, most of the focus on Cooperating Object querying has focused on flat networks. It would be interesting to develop theoretical and empirical frameworks to assess accurately the relative performance of algorithms in flat and multi-tier networks.

In the next years, we also expect some fine-tuning of some of the existing querying paradigms. Among the 5 most important querying paradigms we have: flooding, controlled-flooding (expanding ring search), Random Walks, Location-aided (such as geographic routing), and hierarchical-based. Flooding and Controlled flooding have been studied to exhaustion. Random walks have attracted significant attention but most of the research have been focused on ideal 2-D torii. It is important to design, adjust and assess the performance of random-walk based querying mechanisms according to the real properties of the underlying communication graph. For instance, querying algorithms should consider the degree heterogeneity, link asymmetry and link unreliability or real networks. Location-aided routing has been studied extensively in theory and in practice. Perhaps, a topic where further improvement can be made in this type of querying is to design lighter mechanisms to assign location coordinates. Hierarchical-based querying has not received much attention in Cooperating Object research due to the potentially high costs of maintaining hierarchies with unreliable nodes. Hence, in this domain it would be interesting to investigate light-weight mechanism to maintain a weakly connected hierarchy.

It is also important that researchers try to bridge the gap between empirical and theoretical research. Most of the work in Cooperating Object querying is either empirical (simulation/testbeds) or theoretical. Empirical proposals work on real deployments but the lack formal models do not permit their individually nor cross-layer optimization. Theoretical approaches provide important limits and bounds but they are usually based on unrealistic assumptions which leads to protocols that are not practical useful. Hence, the community should focus in developing querying algorithms with solid empirical and theoretical roots.

6.1.2.6 Data Processing

Section 3.2.6 highlights that there are several existing gaps and emerging research trends in the area of sensor network data processing. These research areas can be classified into three broad strands: Scale, In-network data processing, and Cross layer strategies. The roadmap makes the following observations:

- The next generation of WSN deployments will have large numbers of nodes and data processing techniques have to be tested against this larger scale. In particular, data processing within WSNs must support querying across large scale networks whilst providing timely and accurate responses, regardless of the number of nodes in that network. Hierarchical approaches to data processing architectures in order to achieve this scalability should be explored in this regard.
- Research within in-network data processing at the MAC, WSN and inter-WSN level is still being explored at the individual level and must still be further investigated. One future research avenue within this direction would be to examine the overlap of these techniques, exploring whether optimization techniques appropriate at one level might be appropriate or map to another level.
- Cross layer approaches to data processing are another research gap identified by this roadmap. There are experts within the MAC community, the WSN community and the heterogeneous WSN data processing community working on optimization techniques within their own domain, however it is possible that data processing techniques that are implemented at the MAC layer may not yield a net benefit at the inter-WSN level and vice-versa. Building a model of cross layer data processing will allow optimization techniques to be applied across these multiple layers.
- Query planning is the selection of the optimal query plan to be executed based on some knowledge of the underlying network. Models for cross layer query planning based on the exchange of knowledge between nodes and also between layers should be explored in order to support the previous research area, cross layer query optimization. The execution and enabling of effective cross layer query planning should be a future research direction at all levels within the data processing community.
- Data processing across heterogeneous sensor networks is another emerging research area. There are now several middlewares that abstract from the hardware node level, providing database style abstractions or data stream abstractions to the underlying heterogeneous sensor networks. These approaches however typically hide the complexity and infrastructure of the underlying network thus preventing optimal query planning. Alternative approaches, that reflect the concerns at the individual WSN level, in particular QoS and mobility, have yet to be developed.
- Finally there is an emerging sensor driven web 2.0 community that is try to connect many sensor types including WSNs to the Internet. Semantic approaches to describing both sensors and sensor data in order to enable the visions of "Internet of Things" and "Semantic Reality" are being driven from the top down by this community, yet the intersection between this community and the WSN community is still relatively

weak and should be explored in greater detail. No widely accepted standards are being implemented at either the level of the node or the WSN gateway, nor at the WSN operating system level and at present WSNs and their data streams are being connected to the “Internet of Things” in an ad-hoc manner.

6.1.2.7 Cooperation of vehicles

The scalability is a main gap to achieve efficient cooperation of mobile objects in many scenarios. Most existing methods fail when applied to a large number of mobile Cooperating Objects. This is the case of the methods for optimal coordination, which pose NP hard optimization problems. Existing sensing and perception methods for cooperation also have strong scalability constraints. The same happens when considering routing techniques for large scale scenarios with many nodes.

Real-time is also an identified gap. Many of the above proposed methods have been applied only in simulation without considering real-time properties. Particularly, there are not guarantees to fulfill real-time requirements when considering fast mobile objects and safety constraints. This is particularly true when considering probabilistic methods for sensor data fusion with mobile objects in dynamic environments, as well as active perception and control approaches by using methods such as Partially Observable Markov Decision Processes.

Efficiency in communications is also a gap, particularly when combined with real-time and scalability properties. This is also related to the lack of reliability in communication in uncertain and varying environments.

Network centric real-time actuation, control and decision in mobile object scenarios poses many challenges that imposes a significant departure from the conventional assumptions in Wireless Sensor Networks with static nodes. This also impacts middleware and communication.

Furthermore, position estimation in mobile objects still poses significant problems, particularly in GPS denied environments. In general reliability of the sensors is a main issue in control. Other control-related gaps are virtual sensing and high level interpretation of sensor data, as well as the integration of control with decision and perception involving formal description of the environment and missions

Methods to assure integrity, authenticity and confidentiality are also constrained by the scalability and real-time properties required for the safe cooperation of fast mobile objects.

There is also a need of valid QoS measurements not only for the evaluation of communication and middleware in systems with mobile objects, but also to assess quality in the applications, including for example dynamic coverage with mobile sensors.

6.1.3 Non-functional Properties

6.1.3.1 General aspects

Quality-Of-Service (QoS) issues are the main constituents of Non-Functional Properties (NFP) in Cooperative Objects (CO) systems. Adequate architectural solutions must be found that satisfy the QoS requirements of applications. Some of the relevant QoS aspects in this context are:

- Energy-efficiency / system lifetime
- Reliability / robustness (communication, auto-calibration of sensors, availability, maintainability)
- Timeliness (real-time, traffic differentiation)
- Scalability
- Mobility
- Security
- Heterogeneity
- Cost

Figure 6.3 shows the results of the CONET survey regarding the importance of each non-functional property. As one can see, with exception of heterogeneity and mobility, all the properties where considered of top importance (rank 5) by at least 50% of the interviewees. Furthermore, around 80% of the answers for every property ranked it as at least 4 (exception for mobility which had around 60%). One can also note that there is high interest in reliability/robustness issues since that property had 80% of the answers ranking it as 5, stating its importance for system development.

In addition to functional correctness, computation and communication must be secure and produced "on time" in accordance with application's requirements. It is highly desirable that the energy consumption be minimized. Cooperating Object systems must also be cost-effective, maintainable and scalable. So the "big" challenge is to holistically address all these, often contradictory, QoS attributes.

Another challenge is whether Cooperating Object systems can be based on standardized and/or commercially available (COTS – commercial-off-the-shelf) technologies or if new solutions must be completely designed from scratch. The use of COTS technologies (e.g. communication protocols, operating systems, hardware platforms) might lead to easier, faster and widespread development, deployment and adoption of Cooperating Object systems. Nevertheless, current or even emerging technologies might not be sufficient to fulfill all the stringent requirements imposed by Cooperating Object systems, especially in what concerns the support of non-functional properties.

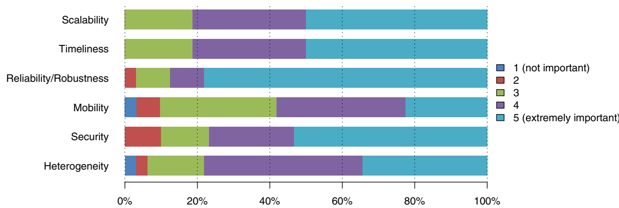

Figure 6.3: *Survey: Non Functional Properties*

6.1.3.2 Scalability

From the very beginning of Wireless Sensor Network and Cooperating Object research, the scientific community has been aware of the importance of building scalable systems. Although there were some research efforts where WSNs of a few hundreds (e.g. [180]) to one thousand nodes ([12]) were deployed, Cooperating Object systems with tens or hundreds of thousands of nodes are still a vision.

Algorithms (e.g. MAC, routing, data processing/aggregation) have been developed to operate as far as possible at different network scales, especially envisaging large scale systems. However, existing approaches are still far away from the desired scalability.

Larger scale may also mean more information sinks, depending on the application. While this can lead to a more complex design and system architecture (e.g. concerning routing), it might also be beneficial in some other perspectives. The existence of multiple geographically distributed sinks might ease the load balancing task, reducing the amount of "bottlenecks" in the WSN. A multiple-tiered architecture may be seen as a particular case of "multiple sinks", since data converges to separate "sink" nodes that may act as gateways to a higher level network.

Dominance-based MAC protocols for WSNs must be further investigated. Computing the MAX and MIN of a certain physical parameter from hundreds or even thousands of sensor nodes, with only one message transaction is very appealing, specially concerning the drastic reduction in time and energy consumption. Other functionalities should also be further explored, such as computing the MEDIAN value, estimating (probabilistic) the number of nodes featuring a certain characteristic (e.g. temperature above a certain threshold, or "I detected toxic leak") or even perform interpolation to get the shape/behavior of a certain physical parameter in space, all these with a time complexity that is independent of the number of nodes [8].

One obvious way for supporting scalability in WSNs is through hierarchical (or tiered) network architectures, such as already stated in 3.3.2. Though eventually leading to more complex network architectures, the multiple-tiered architectural solution that we dubbed "heterogeneous-protocols" seems the most promising for supporting scalability without compromising other QoS metrics (e.g. throughput, delay, reliability). In this case, the communication architecture is composed of a more powerful (e.g. higher energy capacity, radio coverage and bit rate) network technologies serving as a backbone to less powerful (sub)networks at the sensor/actuator level.

Hierarchical network architecture is a well-known and proven principle to make computer networks scale. This type of multiple-tiered network structure brings advantages such as: the communication latency increases very slowly with distance (timeliness), the per-node cost is approximately the same as the per-node cost of the cheapest nodes (cost-efficiency) and it is easy to manage "sleep schedules" for nodes (energy-efficiency).

At the sensor/actuator node level, WSN technologies such as IEEE 802.15.1/Bluetooth (and specially Ultra Low Power (ULP) Bluetooth - WiBree), IEEE 802.15.6 (just formed NOV/2007) – BAN), IEEE 802.15.4 (Physical and Data Link Layers), ZigBee and 6LoWPAN (Network and Application Layers over IEEE 802.15.4), WirelessHART and ISA100 (over IEEE 802.15.4) deserve further investigation. At a higher network level, wireless technologies such as IEEE 802.11/WiFi , IEEE 802.16/WiMAX, IEEE 802.15.3/UWB and GSM/GPRS should be explored. As there might most probably be a need for interconnectivity and interoperability with wired network infrastructures, wired backbone-oriented technologies (such as switched Ethernet (especially Industrial Ethernet – Ethernet/IP, PROFINET, FF-HSE), ATM, FDDI) and sensor/actuator-oriented technologies such as fieldbus networks (e.g. EIB/KNX, LonWorks, HART, ASi, PROFIBUS, Foundation Fieldbus, DeviceNet, ModBus) should also be considered for Cooperating Object systems.

6.1.3.3 Timeliness

As already referred in 3.3.1 and 6.1.3.1, the "big" challenge in large-scale CO systems is to optimize all QoS properties simultaneously, knowing a priori that some (most) of them are contradictory. We can conclude that new design principles are needed in order to engineer large-scale CO systems. Two general principles facilitate the design of large-scale CO systems, particularly in what concerns their timeliness: hierarchical organization of the network and communication-efficient use of the wireless medium.

In what concerns the "timeliness" QoS property of Cooperating Object systems, this "research roadmap" points to the following research directions:

- Explore hierarchical network architectures.
- Investigate how aggregated computations can be used to achieve a time-complexity that is independent of the number of nodes (through prioritized MAC schemes.
- Design protocols and algorithms in an optimized cross-layered approach; analyse

trade-offs in terms of flexibility and interoperability, since the software structure becomes more difficult to update and maintain.

- Build appropriate system planning and network dimensioning tools to be able to achieve optimal timeliness/energy trade-offs.

- Consider timeliness (and real-time) both at the node level (hardware and software) and at the network level; the timing performance of a CO system depends on node hardware design, on the operating system (if any), programming language and style, as well as on the network protocol.

- Investigate existing operating systems (OSs) for resource-constrained embedded systems, specially the most widely used (e.g. TinyOS and Contiki) in a way to incorporate real-time features (pre-emption, priority inheritance mechanism) existing in other OSs (e.g. nano-RK and ERIKA).

- Investigate whether the classical approaches of embedded real-time systems (such as formal WCET analysis, synchronous languages) can be applied to Cooperating Object systems, despite their strong resource limitations, or if more probabilistic-oriented approaches must be followed.

- Continue research on time and energy-efficient routing protocols, particularly trying to merge interesting features from more "mesh-like" (probabilistic MAC/routing, but more flexible, scalable and redundant) and more "clustered-like" approaches (deterministic MAC/routing, but less flexible and redundant, synchronization is complex), to grab the "best of both worlds".

- Design innovative time and energy-efficient mechanisms to mitigate the hidden-node problem.

- Find innovative schemes for MAC in order to improve bandwidth utilization (e.g. avoid "idle/waster" times during nodes power on; using scheduling techniques for sharing guaranteed TDMA slots., e.g. [249]) and for achieving an optimal trade-off between flexibility and complexity in MAC and routing protocols.

6.1.3.4 Reliability/Robustness

As outlined in 3.3.4, Cooperating Object systems hardware must be designed to be resistant to harsh environmental and usage conditions. Moreover, materials and components used in Cooperating Object hardware may harm the flora, fauna or the ecological structure of the environment (e.g. batteries), hence this aspect must be taken into consideration. The increasing tendency for miniaturization, instantiated in technologies such as RFID (Radio-Frequency Identification) MEMS (Microelectromechanical Systems) or SoC/NoC (Systems/Networks on Chip) and for reduction of cost per node should not compromise (or at least at a reduced level) hardware robustness. Actually, the trends for integrating sensing, processing, memory, communication and mechanical functionalities in a single chip may even be explored to improve hardware robustness.

Due to the failure-prone nature of WSNs and since a long system lifetime is usually envisaged, fault-management algorithms must be implemented. Although the techniques enumerated in 3.3.4 are promising in terms of robustness and energy efficiency, further research is needed to address the scalability and network dynamics in designing fault tolerant mechanisms. Some interesting topics to address in the future are:

- When faults occur in WSNs, MAC and routing protocols must accommodate formation of new links and routes to the destination, transport protocols must adaptively decide how to retransmit, and application layer protocols must determine which part of the missing data is critical and what level of loss is tolerable. Therefore, multiple levels of redundancy may be needed and a cross-layer approach exploring the interactions among different protocol layers is desirable.

- None of the mechanisms mentioned in 3.3.4 can recover from all types of faults. There is the need for more robust transport layer solutions that can recover from node failures, link failures and network congestion. Ideally, new methodologies should combine the winning features of existing techniques in an efficient manner.

- The mechanisms presented in 3.3.4 only consider reliability (logical correctness) of data delivery as a performance metric. In fact, timeliness will also be critical for many Cooperating Object applications (refer to 3.3.3). Additional issues to consider are: (1) overallocation of resources (processor/bandwidth) vs. lack of node/network resources in large-scale systems; 2) the trade-offs required to simultaneously support (when required) reliability, timeliness mobility and energy efficiency; and (3) the preferences of applications when all QoS needs cannot be satisfied simultaneously.

- The presence of faults in WSNs introduces uncertainty into standard operations such as answering queries, as data should not be extracted in a purely best-effort manner, but be produced with a clearly defined formal meaning. For instance, it is possible that only a subset of the sensor readings satisfies the application query, thus the network only reports part of the readings filtered by the query. However, the sink does not know whether the remaining reports were not received due to network faults or because results were filtered by the query. If a metric is defined to indicate the completeness of the returned answer, the sink would be better informed. Therefore, it is essential to develop informative quality metrics for sensor applications (network semantics).

Most fault management techniques in WSNs have been integrated with application requirements [345]. Design of a generic fault management technique for WSNs must take into account a wide variety of CO applications with diverse needs, different sources of faults, and various network configurations. In addition, scalability, mobility, and timeliness may also have to be considered.

6.1.3.5 Mobility

Speed, obstacles, radio propagation models, network scale, network density and network partitioning are important factors that must be considered when designing mobility management mechanisms for Cooperating Object systems. Supporting nodes mobility must not impact other QoS metrics in a way that Cooperating Object application requirements are respected.

In many mobility-enabled Cooperating Object systems, the environment is likely to be harsh, leading to unreliable wireless links and therefore constituting an impairment to QoS support, namely to reliability and timeliness. This problem is further complicated when no pre-planned network infrastructure exists. This problem might be overcome by further exploring architectures implementing mobile data collectors (data mules), which collect data from the sensor nodes and deliver it to the sinks. Nonetheless, there are no guarantees on timely data delivery. In contrast, critical applications such as patient monitoring, factory automation or intelligent transportation systems require strict bounds on latency and guaranteed data delivery.

To enable better research, mobility models and benchmarks should be used to evaluate communication protocols and middleware approaches. A framework is required to represent the benchmark datasets so that they can be shared, e.g. for system evaluation and testing. This representation should comprise mobility models derived from real-world data with a combination of some of the following characteristics: user/node mobility, traffic characteristics, network topology, link quality and distribution of nodes - to name a few. Existing simulation tools for WSNs seem to lack mobility support, eventually due to the lack of protocols with mobility support. Cooperating Object systems bring the mobility dimension into the context of WSNs, so future simulators for Cooperating Object systems should encompass mobility support.

The design of a mobility management mechanism fully depends on the existence or not of a localization mechanism (this may impact routing decisions as well). Location information may be precious for better supporting mobility, but may also have a negative impact on network management, energy-efficiency and cost. Location-based routing with geographical coordinates and mobility management has been identified as a potential solution to the issue of communicating data among mobile Cooperating Objects. Such approaches, however, assume that a location service is in place to keep and inform the position of a given node. Some localization services were proposed, but none of them provide a scalable and distributed service. Also, positioning is still inaccurate, so better algorithms are needed. Satellite-based positioning systems (e.g. GPS) are capable of offering a quasi real-time positioning service, but they seem to fail in what concerns cost (in large-scale systems), energy-efficiency (more hardware to feed) and coverage (e.g. in-door and underground). Localization algorithms that are based on transmitting beacon signals often require the utilization of extra hardware and radio channels and the reception of several control packets to reduce the estimation error, which may impact cost and energy-efficiency as well. Therefore, new methodologies for localization must be investigated.

Security may also be affected by mobility: node anonymity and privacy regarding how nodes move in space and time may be required.

Coordination among mobile Cooperating Objects is required for optimal coverage of an area. An important trend is related to the study of how a sensor network can compute, in a distributed way, the path that a mobile CO should follow. This path can be updated depending on the changes of the environment (e.g. mobility of observers, other COs or the phenomenon). More algorithms and theoretical studies are needed in this area.

Routing protocols for WSNs are generally designed for networks that have fixed homogeneous sensor nodes and are based on the assumption that all nodes try to convey data to a central node or one of several backbone nodes. However, in Cooperating Object systems there will be heterogeneous nodes that can be mobile, and the sensed data will be needed by many nodes, i.e., multiple sinks. Generally s peaking, the majority of these algorithms can cope (although not efficiently) with changes of the topology due to node mobility. Most of them, however, react to topology changes by dropping the broken paths and computing new ones, thus resulting in network inaccessibility times that lead to message delays and losses.

Mobility may be particularly difficult to support in cluster-based WSN architectures, due to the cost for maintaining clusters with a set of mobile nodes. Therefore, mobility management mechanisms for cluster-based WSNs must be carefully designed. MAC protocols must also be adaptive to dynamic changes resulting from mobility, as they must transparently re-adapt to node number and density changes.

An efficient mobility management mechanism greatly depends on how far the nodes are able to estimate radio link quality and in a more general way to characterize radio channels. Usually handoff is performed when the current radio link quality is over passed by the link quality of an adjacent cell or cluster. It is thus fundamental for nodes to correctly assess radio link quality. While purely theoretical considerations are straightforward, real experiments lead to a much more complex scenario. Radio links cannot just be identified as "good" or "bad". There is a "transitional region" that can lead to very variable quality and symmetry properties, which is yet to be fully and adequately characterized.

In summary, future research should focus on supporting transparent, seamless, energy-efficient, real-time and reliable mobility management mechanisms in WSNs.

6.1.3.6 Security

The topics addressed in section 3.3.6 have achieved important results but they have not yet reached an adequate level of maturity. The research community has already devoted a great deal of effort to them but more effort is expected in the next future. However, among them a few topics are emerging in terms of importance. One of these topics is low-cost, low-power hardware support to security in Cooperating Objects. So far, most of architectures [531] [222] do not subsume more hardware support than that already provided by communication devices (e.g., CC2420). This choice limits both the performance and the security level of the whole system. Thus, hardware support is necessary to efficiently support cryptographic primitives and, in particular, public key encryption in resource-constrained embedded devices [382]. Public key encryption is an effective cryptographic primitive for key establishment, key distribution and trust management, that are the basic mechanisms for securing spontaneous interaction between cooperating possibly mobile objects [59]. Unfortunately, public key is extremely demanding in terms of computation and thus, so far, has been considered hardly viable for resource constrained devices [353] [425]. Recent works have shown that elliptic curve cryptography is a kind of public key encryption that can be ported on low-end devices [290]. However, engineering cryptography in embedded devices for real world implementation constitute a complex and undoubtedly interdisciplinary research field, involving mathematics and computer science as well as electrical engineering [78]. Alternative choices are available for building an EC-based cryptosystem at different levels that range from the investigation of protocol robustness to software and hardware implementation of the underlying curve and finite fields arithmetic. Even though each of these aspects is often studied in isolation and constitutes a complex subject of interest in itself, the two vertical requirements of implementation efficiency and implementation security make these aspects tightly interdependent.

The rapid growth and pervasive use of Cooperating Objects and their deployment in unattended, often hostile, environments makes it easier for an adversary to gain physical access to these devices to launch attacks and reverse engineer of the system. An adversary can physically manipulate and/or interfere with an object by node capturing, physically tampering with it, and manipulating the object program. A possible response to this kind of attack is program integrity verification, a technique that makes it possible to remotely verify the integrity of the program residing in each device whenever the device joins the network or has experienced a long service blockage. Traditionally, the tamper-proofing of programs or a master secret relies on tamper-resistant hardware. However, this hardware-based protection will likely fail to provide acceptable security and efficiency because strong tamper-resistance is too expensive to be implemented in resource-limited sensor devices, and the tamper-resistant hardware itself is not always absolutely safe due to various tampering techniques such as reverse-engineering on chips, micro probing, glitch and power analysis, and cipher instruction search attacks. Existing approaches to generating tamper-resistant programs without hardware support, namely code obfuscation, result checking, self-decrypting, and self-checking, are unsuitable for sensor networks where a program runs on a slow, less-capable CPU in each sensor device. For these reasons, software-based approaches to program integrity verification have been proposed for sensor networks. These approaches include PIV [347] and SCUBA [416] as remarkable examples. However, these approaches provide security under the assumption of a limited adversary. For instance, PIV assumes that the adversary does not use additional hardware whereas SCUBA assumes that while repairing a compromised node no hardware from the adversary is present. More research is thus necessary to overcome these limitations.

In the case of devices more capable than sensor node, a response to physical manipulation and interference can consist in providing adequate architectural support for securing data and program. Encrypted execution and data (EED) platforms, where instructions and data are stored in encrypted form in memory, while incurring overheads of encryption have proven to be attractive because they offer strong security against information leakage and tampering [515]. These solutions use reconfigurable architectures based field programmable gate array (FPGA) as basic security mechanisms for increased flexibility and performance. However, several attacks are still possible on EED systems when the adversary gains physical access to the system. For example, Gelbart et al. focus on the integrity of the application data to prevent an attacker, who can control the address bus and spoof memory blocks as they are loaded into the processor, from tampering, injecting or replaying the data [144].

Network quality of service (QoS) and network security have been considered as separate entities and research in these areas have largely proceeded independently. However, security impacts overall QoS and it is therefore essential to consider both security and QoS together when designing protocols for ad hoc environments as one impacts the other. The research community has recently acknowledged this gap. ZhengMing et al. propose a mechanism for a distributed dynamic management system which aims to maximize QoS and/or security while maintaining a minimum user acceptable level of QoS and/or security even as network resource availability changes [423]. Pazynyuk et al. are the first who have proposed to discuss QoS problems in WSN by focusing on the availability, reliability and serviceability together as means of providing security integrity in WSN [348].

6.1.3.7 Heterogeneity

As can be inferred from 3.3.7, there is almost no SOTA on Cooperating Object systems heterogeneity, since the number of Cooperating Object system deployments so far is almost insignificant, particularly in what concerns commercial solutions operating in real environments. Therefore, Cooperating Object system's research must start tackling heterogeneity almost from scratch.

As already identified, new classes of resource-constrained embedded system nodes must be clearly identified, defining frontiers between nodes with different characteristics and capabilities. Currently Cooperating Object system nodes span over a large range of types, from MEMS (e.g. for accelerometer), passive RFIDs (e.g. for inventory), active RFIDs (e.g. for toll charge), "general-purpose" motes (e.g. Mica, Telos, FireFly) to more powerful nodes (for routing/gateway and/or processing/control, e.g. iMote, SunSPOT, Stargate). As technology is rapidly evolving in this area, tending for miniaturization, frontiers might turn out to be even harder to define, bringing enormous challenges ahead.

Another challenge is how to tackle the interoperability between sensor/actuator-level communication protocols. From past experience in several fields, it is almost certain that there will be no "single" solution/standard for sensor/actuator-level communication protocol. Wireless protocols such as IEEE 802.15.4 and ZigBee will have to coexist with other emerging ones, such as IEEE 802.15.6 (Body Sensor Networks) , 6LoWPAN, Ultra Low Power Bluetooth (formerly known as WiBree), ISA100 or WirelessHART. Additionally, these wireless protocols for sensor/actuator-level communications will also have to coexist and interoperate with existing wired sensor/actuator networks, such as the ones used in process control and automation industry (e.g. PROFIBUS, ASi, Foundation Fieldbus, HART, DeviceNet, ModBus), in automotive systems (e.g. CAN, FlexRay, TTP, LIN, MOST) and in building automation (e.g. EIB/KNX, LonWorks, HomePlug).

Vertical integration of networks at different hierarchical levels will also be a major challenge. Higher bandwidth and more robust wired (e.g. ATM, Switched Ethernet) and wireless (e.g. WiFi, WiMAX, UWB) communication technologies will have to interoperate with more limited sensor/actuator level networks. Guaranteeing end-to-end QoS brings even more complexity into Cooperating Object system design, i.e. satisfying throughput, delay, reliability, security, energy-efficiency requirements across different hierarchical network levels is not straightforward. Dealing with heterogeneous embedded system nodes hardware/software will not be an easy task. Cooperating Object applications may require sensor/actuator nodes to measure different physical parameters, implying different sensing technology. Also, the same physical quantity may be required to be measured by many sensor nodes (for reliability purposes, or just because there is the need to extract the minimum/average/maximum value of that parameter in a certain region), or even by different types of sensors (for "design diversity" for increased reliability in critical systems or for getting different accuracy levels depending on location and/or time). Both the quantity and diversity of these sensing technologies will bring important challenges (e.g. for hardware design, hardware abstraction layers design, calibration).

As already mentioned heterogeneity in Cooperating Object systems is also instantiated at the operating systems and programming language levels. Operating systems such as TinyOS, Contiki, Mantis, nano-RK, ERIKA have been around for some time, each of them fulfilling specific characteristics. So, it is likely that future Cooperating Object systems (particularly at large-scale) might comprise computing devices running more than one operating systems, leading to additional design complexity. The same applies to programming languages (e.g. nesC, C, Java), simulation/debugging tools, imposing difficult challenges to system designers.

Hosting/client equipment and HMIs are also likely to be quite heterogeneous, in future Cooperating Object systems. Wearable computing equipments are going to be used in a panoply of Cooperating Object applications, for instance HMDs for maintenance in industrial automation or mobile phones in participatory/urban sensing applications. Other equipment, such as database servers, video-surveillance cameras, monitoring/control computers (industrial PCs, PLCs, RCs) mobile robots or transportation vehicles, industrial machinery (welding/painting/assembly robots, machine-tools, roller belts, cranes) will rise the level of heterogeneity to unprecedented levels. Importantly, most Cooperating Object system will support many applications and services, eventually to many different users. Most of these applications and services will impose different QoS requirements, which might dynamically change depending on spatiotemporal issues. This imposes enormous challenges to Cooperating Object system designers, particularly to the ones devoted to networking, since they must properly devise mechanisms such routing/MAC, admission control and scheduling, security, fault-tolerance or data aggregation to encompass such heterogeneous applications and services. The diversity of users that may interact with a Cooperating Object system is also a challenge for system designers, namely in what concerns the diversity in HMIs, safety or security requirements, just as examples. Work on semantics should be further developed so that to ease the Cooperating Object systems users role.

How to support mobility management and fault-tolerant mechanisms, at different levels, is also a major challenge, since Cooperating Object heterogeneity brings additional complexity to the design. Supporting the cooperation between static and mobile objects is not trivial, especially if QoS requirements such as scalability, reliability, timeliness and energy-efficiency must be addressed.

6.1.4 Systems

6.1.4.1 Operating Systems

For sensor networks, mainly two operating systems dominate the market: TinyOS and Contiki . A large community has been built around TinyOS that make it a quasi-standard although it is hardly configurable during run-time and the split-phase programming model used results in complicated programs. The different solutions proposed for these problems show the demand in the community. However, the incorporation of these features would change the TinyOS architecture significantly. Contiki allows multi-threading and dynamic loading by design, making it actually superior. Nevertheless, we think that both worlds will co-exist for a longer time. A recent trend is that operating systems use more and more standard protocol stacks in addition or instead of the previously dominating proprietary protocols. Furthermore, both Contiki and LiteOS offer remote shells that are useful tools for development and configuration. Recent experience with deployments have highlighted the need for efficient operating system support for deployment and debugging as well as the need for self-observation, self-optimization and self-healing of operating systems.

Although many real-time operating systems exist for PCs and also for PDA-sized devices, they are hardly used for very resource-constrained devices like sensor nodes. If control loops are to be pushed more and more down to the actual sensing devices letting them not only be dumb data providers real-time operating systems that target common sensor network platforms have to be developed.

We are see traditional operating systems being scaled down for embedded systems, but the leap to very small devices was not made. Conversely, none of the sensor network operating systems can be extended in such a way that it provides all functionality usually needed in PDA-size operating systems, e.g., GUI support. The extension of the operative range of the operating systems to neighboring device classes should be tackled to allow for easier application development.

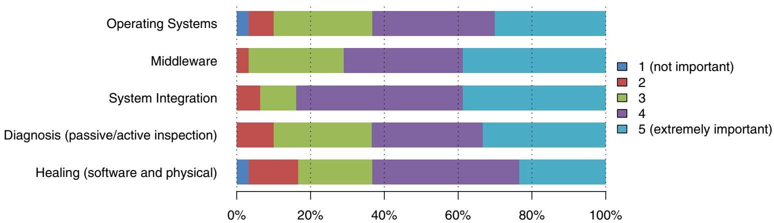

Figure 6.4: *Survey: Systems*

The survey (see Figure 6.4) shows that research on Operating Systems is still considered important, mainly due to real-time issues. Only a minority seems to be satisfied with the existing solutions.

6.1.4.2 Middleware

Since Cooperating Objects are more and more evolving from small experimental to bigger real-world settings it is necessary to reduce the complexity of application programming. For users without experience in distributed systems, middleware that provides a network level abstraction is very promising. However, macroprogramming approaches focus mainly on data stream applications, providing only basic functionality in other aspects that are covered by different middleware approaches. In addition, existing middleware systems fall short of expectations in supporting control applications. However, these are expected to be a key component in the Cooperating Objects world.

Virtual Machines also have the potential to ease the development of Cooperating Object applications significantly, but suffer from runtime overhead. Moreover, no virtual machines exist for all device classes —the gap described for Operating Systems continues here. Research should focus on bringing the formerly separated worlds of sensor networks and pervasive computing together to the Cooperating Objects world.

As described in section 3.4.2, middleware provides very different functionality that help to solve a wide range of common problems. However, no generic mechanism exists for gluing together a complete application using these single solutions. The modular approach of TinyOS can support this task, although the module library is still small and, therefore, code is reinvented again and again but without being compatible to other solutions. An easy construction kit for Cooperating Objects software is desirable, but needs agreements on interfaces for specific problem solutions. Such a construction kit can be part of a graphical user interface that allows to connect the single building blocks, to configure them, e.g., by drawing simple decision trees or by building queries. If the GUI is also used to pre-plan the network of Cooperating Objects this information can be used to assign tasks to the nodes or to name them.

Having clearly defined interfaces of blocks with specific functionality makes adaptation possible. Since the requirements and the environment of applications are dynamic and change often some underlying algorithms have to change as well to behave optimally in the new situation. While sensor networks try to keep the functionality on a node constant and change the behavior of an algorithm, pervasive computing systems redistribute the needed services among the available devices. Both approaches can benefit from each other when combining them in the Cooperating Objects world.

Tightly coupled with adaptive systems is the cross-layer design of algorithms and the system support for such a design. Cross-layer interactions are used widely for optimization purposes or to deal with special properties of Cooperating Objects. To allow for the exchange of an algorithm its data needs to be stored outside of it so that the data is available independently from a specific algorithm.

Further efforts in middleware support for Cooperating Objects are also needed in the area of fault tolerance. Most of the existing systems provides close to no guarantees in case of faults. Nodes running out of battery power, for instance, are eventually recognized and excluded from processing, although no time bounds are provided w.r.t. when this happens. Transient faults, e.g., those arising from sensors temporarily providing erroneous readings are usually not considered. Little or no support is offered to programmers to deal with these situations. High-level programming frameworks where faults are a first-class notion must be designed to develop Cooperating Objects applications. For instance, programming constructs to identify erroneous sensor readings and temporarily exclude a node from the processing may help programmers in improving on the fidelity of data

Finally, most of the existing middleware systems offer little or no support to applications involving mobile sensors. In these scenarios, however, the requirements are typically different from the challenges in static applications. Location is usually of paramount importance, the network topology becomes even more dynamic, and delay-tolerant interactions often become the only way to achieve communication. Programmers are often forced to implement, on a per-application basis, mechanisms such as neighbor discovery and store-and-forward mechanisms. Ideally, middleware systems should be developed to shield programmers from these aspects.

According to the survey (see Figure 6.4) more than 75% of the participiants consider middleware research as important or very important. In contrast to the survey, we rate this topic highest in the "system" domain since we assess the proposed middleware solutions as proof-of-concepts for a specific problem but not as mature, general, adaptable solutions.

6.1.4.3 System Integration

The integration of Cooperating Objects into other systems has to deal with the definition of functionality and their interfaces. When considering sensor networks they are often treated as pure data providers. But to leverage their distributed nature also the control logic and even the actuation should be moved from the central systems to the Cooperating Objects. Of course, this needs new types of middleware to support this new functionality but it changes the fundamental view on the systems as well.

In the current "data provider world", two different approaches exist. The first one concentrates on the network delivering the data and defines an interface for it. These interfaces build on common data query languages like SQL oder XPath. However, to offer additional functionality like event detection the language has to be extended with non-standard constructs. The second approach defines a federation layer with interfaces to the enterprise systems and to the Cooperating Objects world. For large scale deployments this seems to be a powerful approach but its overhead might be too high for small networks.

Since the productive use of Cooperating Objects has just started this research area has not been covered extensively. We expect a set of common functionality and interfaces to appear soon as more and more systems need to be integrated.

The participants of the survey found this topic to be most important (see Figure 6.4), probably due to the fact that not much research has been done so far in this area. Nevertheless, we think that its focus will be on standardization of the interfaces and research challenges are not as manifold as in the other areas of the "system" domain.

6.1.4.4 Debugging and Management Tools

Diagnosis Visibility of the system state is a key prerequisite for diagnosis. It remains a huge challenge to increase the visibility of the system state while minimizing interference with the sensor network. One possible direction to achieve this goal is to consider visibility as a primary goal that drives the design of the system [474] (much like energy efficiency is a goal that drives the design of sensor networks) rather than trying to add visibility to an existing system.

The output of diagnostic tools remains largely disconnected from mechanisms to repair problems. A better integration of these fields of research could lead to solutions where fault detection automatically triggers certain actions to repair the fault.

Work on diagnosis is also largely disconnected from work on programming models and languages. For example, in traditional programming environments, source-level debuggers allow to trace back faults to bugs in the source code of the application. In sensor networks, there is a trend to use high-level languages to simplify sensor-network programming by macroprogramming the whole network rather than programming at the node level. There is currently a lack of source-level debuggers that would allow to trace back faults to certain parts of such a macro program.

At the time of writing, most diagnostic tools can only detect faults that are known in advance. More work is needed on mechanisms that allow to specify or learn a model of the correct behavior of the system and use this model to detect deviations from this correct behavior in deployed systems.

Careful pre-deployment planning and different energy management approaches discussed in this document will help to minimize, but will not completely prevent nodes from failing due to battery depletion. Using on-line battery state monitoring to identify such nodes as early as possible by comparing intended with effective dissipation rates will give the needed time to analyze the existing situation in the network in combination with the modeling used during pre-deployment and to decide for cost-effective actions (e.g. changing the operation mode remotely or actually traveling to the deployment site and add further nodes).

Healing Maintaining and re-establishing network connectivity in both indoor and outdoor environments still represent an open issue that needs to be solved to enable the massive use of mobile platforms in dynamic environments.

Many factories actually rely on autonomous mobile platforms to carry and dispose parts in warehouses. Such vehicles usually embed low-level control (e.g. position, velocity) and range finders relying on communication for high-level control. Deploying an affordable wireless network in a simple or multiple warehouse by adopting redundancy and flexible strategies can be expensive and often unfeasible.

Moreover, in dynamic environments, hot-spot and infrastructure failures are not the main causes of failures. Noise produced by machinery or moving carries, can degrade the quality of wireless connection below acceptable limits in some areas consequently causing “blind zones”. Position, time-duration, and dimensions of such areas relying on many different parameters, such as load composition or machinery working conditions, cannot be forecasted by a central authority and need a wide number of sensors to be detected. A moving vehicle can be easily trapped by one of this holes, reducing the overall performance and forcing the central warehouse authority to employ a technician to rescue the vehicle.

Enabling self-healing on multi-agent systems can drastically increase system performance reducing the effects of these holes by using some devoted or general purpose vehicles as wireless bridge. At present some solution have been proposed to recover static partition by sending a devoted agent as illustrated in previous section, while efforts must be made to use pre-existing vehicles. Recover partitions by modifying the planned trajectory of the working vehicles can reduce the total number of agent, consequently reducing the overall costs, but can compromise the overall performance, e.g. increasing the medium time needed to a vehicle to join its final destination.

6.1.5 Others

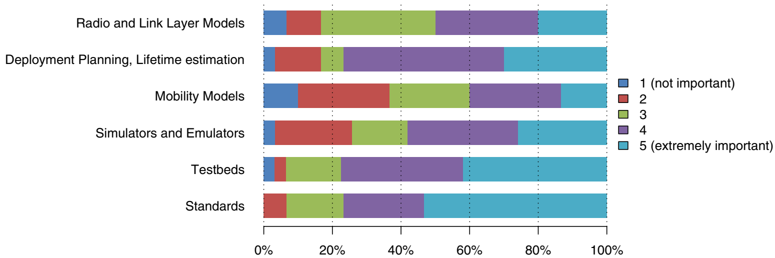

Figure 6.5: *Survey: Other research domains*

6.1.5.1 Modeling and Planning

Interference An important research area concerns the creation of analytical models for the time-variation of interference power, packet loss rates or other transmission quality indicators as mobile BSNs move through different kinds of environments (e.g. urban environments, rural environments, etc.). It can be foreseen that two different types of models are required.

- Synthetic models: here the variations of interference (or packet loss rate) over time are modeled by simple stochastic processes. For the choice of these stochastic processes the aim is not so much to model realistic interference variation behaviors in an extremely realistic manner, but mostly to capture essential properties of these variations and at the same time being analytically tractable.

- Experimental models: these models are based on measured traces and try to capture as well as possible the most important properties of these traces. It is also conceivable to use the traces directly. To be useful for the research community, such traces could be collected on a web server and made publicly available.

These models should include two different types of interferers. On the one hand, a mobile BSN moving through an urban area and operating in an ISM frequency band will pass along several fixed interferers. On the other hand, a mobile BSN might meet other mobile BSNs. The contribution of interference originating from other mobile BSNs is in turn closely related to realistic mobility models for pedestrians or robots.

Radio Link Quality In order to develop a robust and efficient network protocol, it is central to have an accurate representation of the communication graph. However, the myriad of applications and scenarios envisioned for wireless Cooperating Objects (mainly sensor networks) have made difficult to provide a holistic model.

During the last years there have been significant progress in understanding and modeling the unreliable, asymmetric and anisotropic characteristic of low power wireless links. These models have been central in identifying major drawbacks in routing protocols and enhancing their performance, a notable example has been the use of unreliable link models to improve the delivery rate of geographic routing protocols.

Unfortunately, most of these link models focused on benign environments (static environments with good line of sight among the nodes), which do not consider interference effects and temporal dynamics. While there have been some efforts to capture the effects of temporal dynamics and interference on link behavior, the proposed models were developed for specific environments and are difficult to generalize for other scenarios.

Considering the state-of-the-art on pre-deployment tools, there is a real need to develop realistic-yet-general interference and dynamic link models. Similar to what unreliable link models did for geographic routing, these models could be used on the design stage to identify potential risks that could have a severe impact on the overall network performance.

Lifetime estimation Looking at the individual nodes, instead of assuming a constant usage for lifetime prediction, the lifetime issue will more and more be handled in the context of network energy management. However, adapting one of the existing approaches (e.g. [486]) to predict the battery lifetime covering non-linearity and all relevant battery effects to the special hardware requirement of sensor nodes and application requirements of WSNs is still a challenging and valuable task. Also, extending maximization approaches by the special case of WSN-nodes or even incorporating battery effects into (network or MAC) protocol design might give further improvement potential in the node lifetime. Creatively using energy management interfaces [213] from any protocol layer might embody quite some potential for improving network lifetime. Extending such energy management systems by knowledge about the underlying batteries' effects might add further improvement potential.

Network Planning To reduce the deployment cost, it is desirable to have a network planning tool, which assists in achieving efficient Wireless Sensor Network (WSN) deployment. However, because of a great diversity of WSN application-specific requirements and irregularity of communication related issues in WSNs, it is unclear how realistic a generic planning tool can be in constructing a WSN for the real-world applications. Thus, there are still several aspects of WSN planning for future exploration, such as user input for different application scenarios, impacts of environmental constraints, network connectivity and coverage for different topological setups, metrics for evaluating WSN performance, visualization tools for deployment and so on.

6.1.5.2 Testbed and Simulation Platforms

The number of simulators for Cooperating Objects is increasing. Unfortunately, this does not mean that the results are becoming comparable. In most of the cases, the level of detail that is simulated and the underlying assumptions make it impossible to compare the results obtained with different simulators. As standards gain importance, simulators could be used for white-box testing which would enable deterministic interoperability testing with less effort compared to physical meeting for black-box testing. Only recently, simulators that are able to simulate heterogeneous sensor networks consisting of nodes that run different operating systems have been developed. Currently, there is no specification language that enables developers to test the same simulation setups in different simulators. It seems that this would require a common front-end or specification language.

Simulation platforms are essential because developing Cooperating Object applications on the bare hardware is cumbersome. Therefore, it is not surprising that the number of simulators is ever growing. Unfortunately, this diversity is not only a benefit since in general it is not possible to compare the results. Studies have shown large diversities between different simulators in e.g. packet delivery rates, latency when simulating even simple scenarios. Hence, there is a strong need for making simulation results comparable across different simulation platforms. One solution to this problem might be common front-ends or new simulation specifications languages.

Being able to simulate different operating systems and platforms within the same simulation does not only enable comparable results but also interoperability testing. The latter is of major importance as standards for Wireless Sensor Networks are emerging,. We have seen first steps towards that direction but definitely more work is needed here.

Simulators support rapid application development but for testing real application behavior they are in most cases less realistic than experiments done within a controlled testbed. So-called hybrid simulation extends the simulation environment with nodes from real testbeds. This approach is in particular useful for studying scalability issues. Unfortunately, a hybrid simulation environment is not deterministic and hence not repeatable. An alternative approach is sensor network check-pointing that transfers the state of the sensor network between simulator and testbed.

6.1.5.3 Standards

Some year ago Bluetooth and ZigBee were among the few existing standards for Cooperating Object communication. While at that time many expected ZigBee to be used in all application domains, we currently see a strong position of ZigBee in the area of building automation. As shown in Section 3.5.3, there is now a much larger and diverse number of standards for Cooperating Objects. In particular, IP e.g. 6LoWPAN is now a viable alternative for Cooperating Objects and it would not be surprising if other standards will embrace IP. Also, it remains to be seen whether WirelessHART and ISA100 will merge. ZigBee's application profiles might be of interest for other standards as well.

Due to the number of emerging standards, the question of co-existence of these standards needs to be addressed. Standards must be aware that other standards and protocols use the same frequency bands. For some of the standards, only proprietary protocol implementation exist. Therefore, open source implementations of the standards is required. There is also a need for tools for interoperability testing. In particular, the development of simulators that enable interoperability testing for different platforms must be considered as extremely useful.

6.2 Timeline

The gaps described in the last sections identify the type of work that needs to be done in certain areas, but do not provide any indication as to when they are expected to be solved. Therefore, we asked ourselves and the participants of our survey to estimate when each gap will be solved. It is important to mention that these estimations assume that research is driven by the current needs of the users and that no research direction is promoted due to, for example, this roadmap. Thus, the timeline does not show the ideal state. But this process made it possible to extract the predominant research areas afterwards.

Let us now look at each one of the groups of gaps mentioned in previous sections in more detail.

6.2.1 Hardware

We expect Sensor Calibration to be solved relatively soon in comparison to other gaps because unless this issue is solved in a satisfactory way, it is hard that sensors can be used in environments where costs play a major role, such as in the Home and Office domain. Other more industrial domains are willing to pay higher prices and, therefore, more sophisticated methods for sensor calibration can be used.

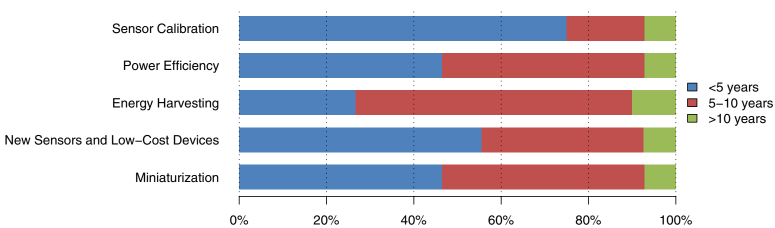

Figure 6.6: *Survey: Hardware*

A similar argumentation can be used with the Power Efficiency gap. We expect a major breakthrough in a short to medium term, because of the importance of this issue for the adoption of technology. It is clear that this will remain an issue that needs to be investigated further in the future, but unless we are able to provide good solutions in a relatively short period of time, there might be a decrease in the interest of Cooperating Object technology.

Energy Harvesting , on the other hand, is a very hard problem that will require more time to find solutions that could be used on a more widely basis. It seems that the need for more efficient materials that transform other types of energy into electrical impulses will remain an issue in the long term. Here, the CONET consortium is a bit more pessimistic than the survey participants who expect Energy Harvesting to be solved in 5 to 10 years.

The same happens with the New Sensor and Low-Cost Devices gap. There are quite a few research groups, research institutions and companies trying to produce cheaper and cheaper devices, and this will continue throughout the evaluated period since the need for new and improved devices will increase with the adoption of technology. The addition of new application domains implies not only the creation of new devices, but also changes in the requirements of already existing ones that will have to be improved accordingly.

Finally, miniaturization goes hand-in-hand with cost and power-efficiency and we believe that the need for smaller and smaller devices will continue throughout the studied period.

In the end, there might be even a fusion of different research areas, such as nanotechnology, that attempt to create even smaller devices than the ones envisioned for the field of Cooperating Objects.

6.2.2 Algorithms

The next key challenge in Cooperating Objects research is scalability. Taking as an example the area of wireless sensor networks, we observe that in 2002 the largest test-bed consisted of around 160 nodes (Intel Research Labs, Berkeley). In 2004, the Extreme Scale Wireless Sensor Networking test-bed had around 1000 nodes (Ohio State University, USA). Nowadays (2009), there are an increasing number of groups with test-beds consisting of nodes in the order of a few hundreds.

The increasing availability of hardware has guided the design and validation of algorithms *from simulation-based to small-scale test-beds*. In the next years, we need to bridge the gap between small-scale and large-scale test-beds.

Considering that reliable and inexpensive hardware is a necessary condition to deploy large scale networks, the timeline of Algorithms is tightly coupled with the timeline of Hardware. In the short term, we believe that a few large-scale test-beds will appear. These large test-beds will allow the research community to filter the best candidates among the myriad of Localization, MAC, Querying and Routing algorithms proposed up to date. Also, these large test-beds will permit to fine-tune the reliability, robustness and energy

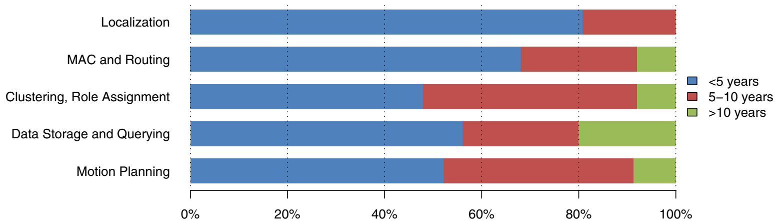

Figure 6.7: *Survey: Algorithms*

efficiency of the most promising algorithms.

The areas of Localization and MAC and Routing have received significant attention in the last years. Hence, we expect these areas to develop fully in the near term. For Clustering and Motion Planning , we see solutions in short to medium term, the latter are especially due to the combination of mature robotics techniques with Wireless Sensor Network technology. In contrast, Data Storage will remain a medium to long term research area since new applications will require new data storage and querying techniques, for example, for large-scale or heterogeneous networks.

6.2.3 Non-functional Properties

The non-functional properties (NFPs) are considered to be of paramount importance for Cooperating Objects (COs) systems. This is reflected by the market analysis presented in chapter 5 and also by the answers to CONET survey (section 6.1.3.1). As already referred reliability/robustness, heterogeneity, timeliness, security, heterogeneity and mobility are quality-of-service (QoS) properties that must be observed in all Cooperating Object systems and fulfilled for each particular application in both individual and integrated perspectives. Mobility is probably the only exception in what concerns the particularities of each Cooperating Object system, in the sense that only a subset of the applications may require mobility support.

The current state-of-the-art and state-of-technology reveals a strong immatureness and a clear lack of solutions (protocols, software/hardware architectures, technology) in respect to these NFPs. Current real-world applications and even research-oriented test-beds exist in a relatively small number and feature just up to some hundreds of sensor/actuator nodes. Market studies (e.g. ON World Inc.) forecast mass deployments of Cooperating Object systems (sensor/actuator networks, pervasive Internet, smart environments) at a global scale, but this seems to be a vision that will see the light only in more than one decade.

Research on improving the timeliness , security and reliability/robustness of Cooperating Object systems are still at a very early stage, particularly for the latter. Scalability is being considered by researchers (e.g. algorithms, methodologies, protocols), but results are still either incomplete, immature and/or yet to be validated in real-world applications. Almost no work exists on supporting mobility (nodes, node clusters) in Cooperating Object systems. While successful results are not obtained using homogeneous Cooperating Object systems, it will be hard (almost impossible) to support high levels of heterogeneity, such as the coexistence and interoperability between different hardware platforms, network protocols, operating systems, middleware and applications.

Even more difficult is to fulfill and balance all these NFP/QoS properties at the same time, i.e. in a holistic perspective, since most of them are contradictory (i.e. improving one of them may harm the others). While a minimum level of maturity in each NFP must be reached, a bigger challenge is to devise system/network dimensioning methodologies and tools that are able to support system designers on balancing these properties in a way that system/application requirements are met. This is why we preclude that mature solutions to fulfill these QoS properties in a holistic fashion will only be achieved in a decade or so.

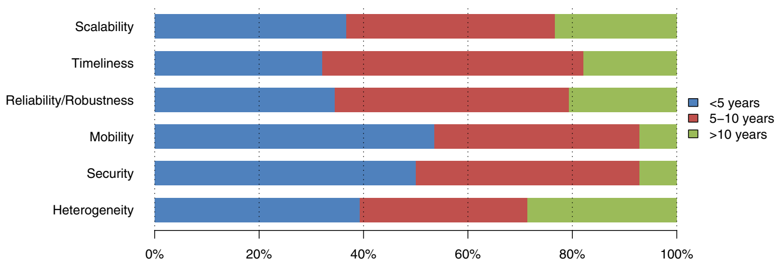

Figure 6.8: *Survey: Non Functional Properties*

Regarding the result of the CONET survey (Figure 6.8), there is a slight optimism in the sense that roughly 40% of the respondents expect some concrete results to be achieved in the short-term. Other 40% believe that achievements will only appear in the medium-term, while the remaining 20% are more defensive when stating that significant solutions on these QoS properties will only see the light in the long-term.

6.2.4 Systems

We expect the issues related to Operating Systems to be solved soon since they are the basis for all Cooperating Objects software. This is confirmed by the results of the survey (Figure 6.9). Since integration of Cooperating Objects into other systems will increase in the next years solutions for System Integration will emerge soon and (de-facto) standards will be established.

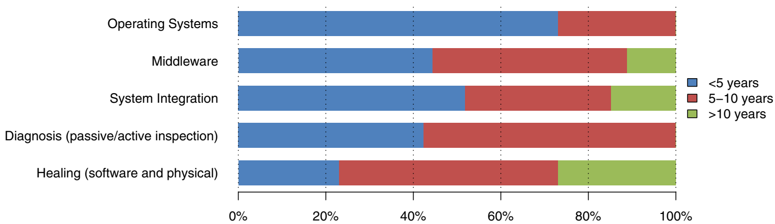

Figure 6.9: *Survey: Systems*

On the other hand, Middleware and Programming Models will take a longer time to be solved. A challenging task will be the determination of the right abstraction for a wide range of applications. It is unclear how concurring abstractions can be evaluated in a quantitative way. For application developers, middleware operation must be predictable. Therefore, the systems have to be clearly documented and the implementations have to be well-engineered.

Adaptive Systems , that is, middleware solutions that use the underlying operating system and programming abstractions to provide additional functionalities that deal with adaptation, are expected to remain a hot topic of research throughout the studied period. The reason for this is the need for optimization of applications independently of the environment they are immersed in. This can be achieved by the use of adaptive systems that need to be worked on in order to be able to cope with new application domains that will arise as a result of a wider adoption of Cooperating Object technology.

While we believe that diagnosis and healing will remain a relevant issue in the medium and long term, respectively, it can be expected that certain aspects of this problem domain will be solved in the near future. The reason for this is that many problems result from subtle bugs in the system software and communication protocols. Due to a lack of established standards, there is a tendency to develop custom solutions, which are often not sufficiently tested under real-world conditions. Once appropriate standards are adopted, we can expect them to be thoroughly tested by the community. However, it is often argued that sensor networks necessarily require application-specific hardware and software solutions (e.g., due to constrained resources) which implies that every deployed system continues to include at least some components that have been specifically developed for a specific application and which have been tested to a lesser degree.

Another aspect is that current hardware platforms are primarily made for research and experimentation, not for production use. Once industry picks up Cooperating Objects at a larger scale, we can expect that more reliable platforms become available. Nonetheless, Cooperating Objects will remain large and complex distributed systems that are exposed to highly dynamic and unpredictable environments, making (partial) failures a common event that has to be dealt with. To this end, we need to develop solutions that are designed to repair most of these problems without involving the user, and provide effective tools to support the user in dealing the remaining problems.

6.2.5 Others

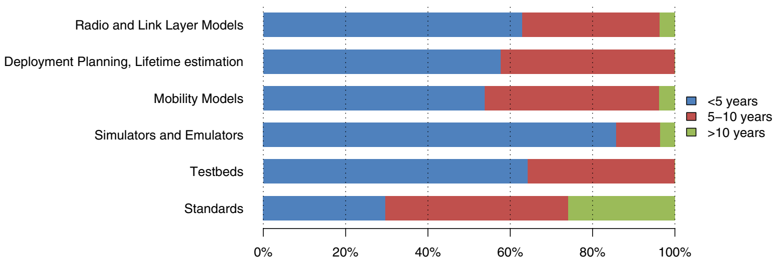

Figure 6.10: *Survey: Other*

It will take more time to develop realistic radio and link layers models as well as mobility models . The latter require large collections of real-world traces. While there exist radio and link layer models, many of them are not that satisfying. Furthermore, with new radio technology including for example the emerging cognitive radios, new radio and also link layer models will need to be developed.

These models, together with realistic battery and power consumption models, are required to develop accurate Deployment Planning and Lifetime Estimation tools. Therefore, we believe this area to get solved in medium term, which is a bit more pessimistic than the survey result.

As stated earlier in Section 6.2.2 there exists a number of large-scale sensor networks testbeds. It will take some time and effort to adapt these to a larger range of Cooperating Objectsbut from a technical perspective we see no major barriers.

There already exit a huge number of Cooperating Object simulators. Recent work has shown the usefulness of emulators for e.g. interoperability testing. A problem is that while a number of useful features exist, these are spread over a large number of different simulators. Finding good ways of merging simulator features or simulators or incorporating different features simulators into one or more simulators is hence a major issue that requires effort and time as well as new methods and insights.

Standardization is a long and difficult process. Industry and users have to agree on a common technology and algorithms. Therefore, we see this as a long term issue. Fortunately, the Cooperating Objects community is adopting existing standards, e.g. ZigBee, but still deviations are used. With more real-world installations, experience will show which technology is superior, which will help in the selection for a standardization basis.

6.3 Conclusions

There are still many open and challenging gaps in Cooperating Objects research that are worth pursuing. Although we do not claim to have mentioned all relevant gaps, the ones presented in this chapter are the ones that have been detected by our experts and the ones surveyed at different events. In general, both groups of experts coincide in the importance of the areas and mostly in the ones that require more attention in the next years.

Regarding the timeline presented for each one of the areas, a general consensus also seems to exist among experts. Their estimation regarding the time when these problems will be solved from the point of view of research has helped us in determining, together with the information about importance of gaps, the predominant work areas presented in the following chapter.

Chapter 7

Predominant Work Areas

After the comprehensible overview of the the research gaps, timeline for their development, market predictions and potential inhibitors, we proceed to present the predominant work areas and the key issues that should be addressed in the next years.

The CONET consortium offers a good mix of academia and industry and contains the major players in both the industrial and academic areas, either as part of the core team or in the advisory boards. For the analysis of gaps, we have included their opinion, as well as the view of manufacturers and product developers that do not deal with Cooperating Objects technology on a daily basis.

As one could expect there are some differences with respect to the academic and industrial view e.g. concerning the actual algorithms that make up an application. Thus future research will have to be coordinated, address application-driven issues and mainly focus on solving real world problems in order to avoid diverging too much from the expectations of industry and enhance existing or offer new services to the citizens and businesses.

The key issues that need to be tackled have been determined by selecting the gaps that will require a significant amount of work to solve and that are also seen as roadblocks for other related activities. The selection of the areas also took into account the market potential of different application areas and the market entry inhibitors identified by the studies, our own survey and several discussions with key industrial players.

Although the gaps identified in the previous chapter still hold and need to be solved, the topics shown in this chapter should receive the most attention in the following years in order to advance the area of Cooperating Objects in the most effective way.

7.1 Energy Considerations

For small, embedded or long-running Cooperating Objects power sources and power consumption are crucial. The miniaturization of batteries does not experience the same growth in storage than the processing power or the miniaturization of devices (that follows Moore's law). The specific characteristics of small Cooperating Objects have to be taken into account in battery technology research: It is not necessary to draw high current from the power supplies, but they have to deliver low power over a long period of time.

If energy cannot be stored for the intended lifetime of the Cooperating Object, energy needs to be harvested. The available technologies have been presented in Section 3.1.3. Most approaches are not ready yet to meet the needs of Cooperating Objects technology.

On the other hand, Cooperating Objects need to be built in a power efficient way, both for hardware and software. As we pointed out in Section 6.1.1.1 the integration of the building blocks of mote, their selective activation or deactivation and the use of data signal processors can reduce the power consumption. Care must be taken that these approaches do not lead to very specialized hardware that is not useful for a wider range of experimental and proof-of-concept installations.

The third pillar in this work area are power efficient algorithms. Although energy was one of the major optimization goals since the beginning of Wireless Sensor Networks, new issues arise due to the platforms mentioned before and due to the collaboration of different algorithms. Cross-layer optimizations are proposed to tackle this, but they are not used throughout and consistent. General software support for cross-layer interactions is still marginal.

It is obvious that this work area needs the collaboration of different researchers since the subareas are strongly interacting and advances in one subarea opens new challenges but also chances on other subareas.

7.2 Localization

Although localization is already in the focus of many researchers of the Cooperating Object community, it is an important research area. In outdoor scenarios GPS can be used, which delivers an accuracy of approx. 15 meters. For some applications, this is not sufficient, for others GPS is not an option at all. Moreover, GPS is expensive, also in terms of energy.

For indoor use, the localization problem is not solved to the level of satisfaction required by most applications. Good solutions should be quick and easy to install and provide a high accuracy, not only at room level but in the range of a few centimeters. With the availability of this technology, many promising in-house applications will profit from such techniques or will become possible at all.

Another open problem is the seamless transition of location information while transitioning from an indoor to an outdoor area and vice versa. Nowadays it is not possible to have a system that can be used inside and outside of buildings with enough accuracy. The integration and combination of different localization approaches is, therefore, needed.

7.3 Data Management

The handling of large amounts of collected data is a promising research area, including distributed multi-sensor environment perception. This includes a variety of single techniques that should be combined in the appropriate way to create optimized solutions. For example, data has to be aggregated in order to reduce the amount of data to be transmitted. At the same time, its storage within the network is not an easy task that should be addressed, as well as the methods to search and query it. To improve on the availability of data, it is possible to use replication techniques, but this might lead to consistency problems that should be solved in an energy-efficient way.

The single data management techniques influence different layers. Due to the often large amount of data they will profit enormously from cross-layer optimizations and, therefore, contribute to the Power Efficiency goal as well. These optimizations take part not only on the nodes processing the data, but also on the nodes issuing the query and, thus, planning their execution.

Since Cooperating Objects networks are more heterogeneous than Wireless Sensor Networks where many Data Management approaches have been developed for this new characteristic has to be taken into account to optimize planning, processing, storage or retrieval.

In general, the goal of the data management techniques is to simplify the usage of Cooperating Objects. Especially manufacturers are concerned about the integration of Cooperating Objects into existing software. Therefore, they need an easy and intuitive interface to get the interesting data without knowing the internals of a Cooperating Object.

7.4 Non-functional Properties

Non-functional properties are not directly implemented as algorithms but influence their design. Many algorithms have been developed that are said to be scalable and to be able to work well in mobile environments. However, there is no real classification of scalability and mobility. Thus, one scalable algorithm supports only a few hundred nodes while the other works with tens of thousands. Moreover, a single application area might differ significantly from another one and its requirements regarding non-functional properties might not hold in the new environment. Therefore, algorithms should concentrate on a few properties that are motivated well by real application scenarios.

The same also applies to Quality of Service concepts. The difference to the aforementioned properties is that they are given by the environment while the required quality of service is given by the user and can change over time. A detailed study of the application areas about these requirements is missing. Most of the time, developers tend to design an algorithm which gives "best" quality of service while neglecting that it can be useful to reduce the quality of service requirements to improve other properties, e.g. power consumption. The importance of Quality of Service concepts is also underlined by our survey (see Section 5.4). "Confidence in technology" is very important for the adoption of Cooperating Objects. This includes all "soft" factors, e.g. that systems are reliable, that they deliver their data in time, that they are not tampered with.

"Security" as a property can be considered a type of quality of service, but should be considered explicitly. All application scenarios will require a certain level of security if they evolve from the prototype state. For control applications, it is obvious that an intruder can cause damage to the controlled objects. But also for simple monitoring applications, a secure network will increase the trust of people in technology. Since an existing system cannot be secured afterwards but has to be designed with security in mind – which is usually not the case –, new research projects have to consider security aspects from the very beginning as a cross-layer task.

7.5 System Support

The vision of Cooperating Objects that are fast to develop, easy to use and maintain can become reality if appropriate system support is available. Different middleware solutions must be combined to form a construction kit with simple building blocks that can be just plugged together.

Due to the dynamic nature of Cooperating Objects adaptation support should be an integral part of such a construction kit. In the best cases, the end user is not even aware that adaptation is happening and it is not necessary to specify in what cases which algorithm should be used. Rather, the end user can describe in a high-level manner how the properties are linked and the underlying system takes care of the rest.

Although adaptation is a powerful concept it is not a magic bullet. Therefore, good diagnostic approaches are needed that should be able to determine its normal behavior and deviations from it. Tightly coupled with deployment planning tools they can give hints where and how to improve an installed network or where proactive maintenance is advised to prevent more severe network conditions.

Such improvement or maintenance task can be carried out automatically by mobile platforms that temporarily or permanently act as additional network nodes or are even able to replace or recharge nodes.

7.6 Modeling and Planning

Real-world experiences usually show a big gap between the envisioned and actual network behavior. All tools that are used before deploying a network, i.e. simulators or planning tools, rely on the validity of the underlying models. Many different models are needed to fully cover a network of Cooperating Objects: accurate radio models , especially for indoor scenarios, sensor models to assess the sensor coverage of the monitored area, mobility models for mobile Cooperating Objects or mobile targets that are to be tracked, energy models for power supply and power consumption , event models that exhibit the patterns of the phenomenon that is to be monitored.

It is important that these models influence each other. For example, mobile objects or certain events disturb radio connectivity or obstruct sensors. Therefore, models need to take into account the current state of other models.

Planning tools should select the appropriate models for a specific scenario. Since models are usually general they need to be parameterized for this scenario. For example, a planning tool could suggest positions for a number of selective radio link measurements that allow to adjust the radio model to the actual characteristics of the planned deployment. For deployed networks, results of the diagnosis tools can help to tune the radio models again to further improve the networks by iterative planning.

7.7 Simulators and Testbeds

Simulators and testbeds are an important instruments for research on Cooperating Objects. Therefore, a variety of them already exist. Unfortunately, results of different simulators are not comparable right now. An interesting question is if equivalent classes of simulators can be defined, what the distinguishing factors are, how they can be formalized, and if this formalization can help to prove the equivalence of results obtained by other simulators.

The integration of simulators and testbeds is another important area to support testing real-world application behavior for a large number of nodes. Different approaches are possible: the testbed can be embedded in the bigger simulated scenario and messages and events are converted between both worlds, or specific situations of the simulation could be executed in the testbed and the feedback is taken to adjust the models controlling the simulation.

To have to possibility to test an application with different testbeds standards for interfaces are needed. This would also allow for the combination of multiple testbeds for a single application test to test heterogeneous environments. Similar to the integration of simulators and testbeds the transformation across the borders needs to be solved.

7.8 Standardization

After a phase of research where different hardware platforms, radio stacks, and protocols have been proposed, there is a need for a consolidation and standardization phase that drives the adoption of this new technology.

When discussing this issue with the industry, many manufacturers are still waiting for standards and standardization committees to indicate which technologies should be considered in more detail and, in a sense, show them the way that will be favored by most competitors and potential suppliers. This also affects their prices and will definitely benefit from hardware and software standards. On the other hand, standardization committees composed by the major industry players should be formed in order to promote this technology. One of the first attempts is the standardization of ZigBee, but other issues, not just communication should also be addressed. For example, IEEE 1451 should be extended and adapted to Cooperating Objects.

Developers and researchers can also benefit from the use of standards, since they can concentrate on only a few platforms, communication technologies, operating systems and algorithms. Therefore, the development of standard platforms, software, interfaces and tools for debugging and development should be favored over the development of more proprietary forms of hardware and software.

7.9 Conclusions

In all domains of Cooperating Objects research areas have been identified that need to be reinforced since their solution is vital for the adoption of Cooperating Objects. Many proposed predominant work areas do not only cover a single topic but present different and interdependent domains. Strong collaboration between different researchers in different domains is, therefore, necessary to tackle these complex tasks.

Chapter 8

Conclusions

It is clear that the field of Cooperating Objects, is a very dynamic one that has the potential of drastically changing the way people interact with the physical world as well as how business systems integrate it in their processes. We are still at the dawn of an era, where a new breed of applications and services, strongly coupled with our everyday environment will revolutionize our lives even in a deeper way than the Internet has done in these past years.

In this research roadmap, we have tried to provide a thorough overview of the current status as well as the possible directions that research in this domain might take in the next 7 to 15 years. We have presented promising applications domains, and given a glimpse of effect that Cooperating Objects might impose upon them.

We have also given information about a probable timeline for the development of research and have attempted to characterize the points in time where major breakthroughs will allow for Cooperating Objects technology to become mainstream. Additionally, we have tried to pinpoint major inhibitors and potential roadblocks and given concrete suggestions to avoid them.

It is clear that such a disruptive approach, can not be managed by a single actor nor be fully exploited, without coordinated cooperation among researchers, industrial partners, end-users, financing institutions, policy regulators etc.

It is our intention to keep this roadmap updated in the next years, and incorporate multi-domain feedback from diverse actors e.g. industry, academia, etc To that sense this is an open invitation to all readers to provide us with visionary insights as well as pinpoint to roadblocks and future research directions.

Bibliography

[1] Ameer Ahmed Abbasi and Mohamed Younis. A survey on clustering algorithms for wireless sensor networks. *Comput. Commun.*, 30(14-15):2826–2841, 2007.

[2] T. Abdelzaher, B. Blum, D. Evans, J. George, S. George, L. Gu, T. He, C. Huang, P. Nagaraddi, S. Son, P. Sorokin, J. Stankovic, and A. Wood. EnviroTrack: Towards an environmental computing paradigm for distributed sensor networks. In *Proceedings of the 24th IEEE International Conference on Distributed Computing Systems (ICDCS'04)*, March 2004.

[3] Tarek Abdelzaher, K. Shashi Prabh, and Raghu Kiran. On real-time capacity limits of multihop wireless sensor networks. In *RTSS' 04: Proc. of the 25th IEEE Real-Time Systems Symposium*. IEEE Press, Los Alamitos, CA, 2004.

[4] K. Aberer, M. Hauswirth, and A. Salehi. Infrastructure for data processing in large-scale interconnected sensor networks. In *MDM 2007*, 2008.

[5] Karl Aberer, Manfred Hauswirth, and Ali Salehi. A middleware for fast and flexible sensor network deployment. In *Proceedings of the 32nd international conference on Very large data bases (VLDB '06)*, pages 1199–1202. VLDB Endowment, 2006.

[6] Advisory Council for Aeronautics Research in Europe. Strategic research agenda (acare). Technical report, Advisory Council for Aeronautics Research in Europe (ACARE), 2004.

[7] A.D. Amis, R. Prakash, T.H.P. Vuong, and D.T. Huynh. Max-min d-cluster formation in wireless ad hoc networks. *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 1:32–41 vol.1, 2000.

[8] B. Andersson, N. Pereira, W. Elmenreich, E. Tovar, F. Pacheco, and N. Cruz. A scalable and efficient approach for obtaining measurements in CAN-Based control systems. *IEEE Transactions on Industrial Informatics*, 4(2):80–91, 2008.

[9] B. Andersson, N. Pereira, and E. Tovar. Exploiting a prioritized MAC protocol to efficiently compute min and max in multihop networks. In *Proceedings of Fifth Workshop on Intelligent Solutions in Embedded Systems*, Madrid, Spain, June 2007.

[10] Panayiotis Andreou, Demetrios Zeinalipour-Yazti, Panos K. Chrysanthis, and George Samaras. Workload-aware query routing trees in wireless sensor networks. In *MDM '08: Proceedings of the The Ninth International Conference on Mobile Data Management (mdm 2008)*, pages 189–196, Washington, DC, USA, 2008. IEEE Computer Society.

[11] A. Arora, E. Ertin, R. Ramnath, M. Nesterenko, and W. Leal. Kansei: a high-fidelity sensing testbed. *Internet Computing, IEEE*, 10(2):35–47, March-April 2006.

[12] A. Arora, R. Ramnath, E. Ertin, P. Sinha, S. Bapat, V. Naik, V. Kulathumani, Hongwei Zhang, Hui Cao, M. Sridharan, S. Kumar, N. Seddon, C. Anderson, T. Herman, N. Trivedi, M. Nesterenko, R. Shah, S. Kulkami, M. Aramugam, Limin Wang, M. Gouda, Young ri Choi, D. Culler, P. Dutta, C. Sharp, G. Tolle, M. Grimmer, B. Ferriera, and K. Parker. Exscal: Elements of an extreme scale wireless sensor network. In *Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA '05)*, pages 102–108, Aug. 2005.

[13] A. K. Arumugam, A. Doufexi, A. R. Nix, and P. N. Fletcher. An Investigation of the Coexistence of 802.11g WLAN and High Data Rate Bluetooth Enabled Consumer Electronic Devices in Indoor Home and Office Environments. *IEEE Transactions on Consumer Electronics*, 49(3):587–596, August 2003.

[14] Atmel Corporation. Atmel AVR 8-Bit RISC. http://atmel.com/products/avr/default.asp. [Online; accessed: 14.05.2009].

[15] Aubade. IST Project: A Wearable EMG AUgmentation system for roBust behAvioural understanding. http://www.ist-world.org/ProjectDetails.aspx?ProjectId=9ca0a79cc25642879c4a0daec0b7a0ec. [Online; accessed 14.05.2009].

[16] Various authors. Special issue on cognitive wireless networks. *IEEE Wireless Communications*, 14(4):4–52, August 2007.

[17] Chen Avin and Carlos Brito. Efficient and robust query processing in dynamic environments using random walk techniques. In *IPSN '04: Proceedings of the third international symposium on Information processing in sensor networks*, pages 277–286, New York, NY, USA, 2004. ACM.

[18] Chen Avin and Bhaskar Krishnamachari. The power of choice in random walks: an empirical study. In *MSWiM '06: Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*, pages 219–228, New York, NY, USA, 2006. ACM.

[19] Aware project. http://www.aware-project.net. [Online; accessed 14.05.2009].

[20] Aware home research initiative. http://awarehome.imtc.gatech.edu/. [Online; accessed 14.05.2009].

[21] Stefan Axelsson. Intrusion detection systems: A survey and taxonomy. Technical Report 99-15, Chalmers Univ., March 2000.

[22] Fan Bai and Ahmed Helmy. A survey of mobility models, 2004. http://nile.usc.edu/~helmy/important/Modified-Chapter1-5-30-04.pdf.

[23] Fan Bai, Narayanan Sadagopan, and Ahmed Helmy. IMPORTANT: A framework to systematically analyze the Impact of Mobility on Performance of RouTing protocols for Adhoc NeTworks. In *Proc. of the IEEE Infocom*, pages 825–835, 2003.

[24] Xiaole Bai, Santosh Kumar, Dong Xuan, Ziqiu Yun, and Ten H. Lai. Deploying wireless sensors to achieve both coverage and connectivity. In *MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 131–142, New York, NY, USA, 2006. ACM.

[25] Yuebin Bai, Jinghao Li, Qingmian Han, Yujun Chen, and Depei Qian. Research on planning and deployment platform for wireless sensor networks. In *Proceedings of the Second International Conference on Advances in Grid and Pervasive Computing, GPC '07, Paris, France, May 2-4, 2007*, pages 738–743. Springer Berlin / Heidelberg, 2007.

[26] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla. GloMoSim: A Scalable Network Simulation Environment. *UCLA Computer Science Department Technical Report*, 990027, 1999.

[27] Body area network monitors arousal level. http://www.holstcentre.com/NewsPress/PressList/Body%20area%20network%%20monitors%20arousal%20level.aspx. [Online; accessed 14.05.2009].

[28] J. Barton and V. Vijayaraghavan. Ubiwise, a simulator for ubiquitous computing systems design. Technical Report HPL-2003-93, HP, 2003.

[29] Stefano Basagni. Distributed clustering for ad hoc networks. In *ISPAN '99: Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '99)*, page 310, Washington, DC, USA, 1999. IEEE Computer Society.

[30] Christian Becker, Marcus Handte, Gregor Schiele, and Kurt Rothermel. PCOM - A Component System for Pervasive Computing. In *Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04)*, pages 67–76, Washington, DC, USA, 2004. IEEE Computer Society.

[31] Christian Becker, Gregor Schiele, Holger Gubbels, and Kurt Rothermel. Base - a micro-broker-based middleware for pervasive computing. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom '03)*, pages 443–451, Washington, DC, USA, 2003. IEEE Computer Society.

[32] Christian Bettstetter. Mobility modeling in wireless networks: categorization, smooth movement, and border effects. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(3):55–66, 2001.

[33] Shah Bhatti, James Carlson, Hui Da, Jing Deng, Jeff Rose, Anmol Sheth, Brian Shucker, Charles Gruenwald, Adam Torgerson, and Richard Han. MANTIS OS: An embedded multithreaded operating system for wireless micro sensor platforms. *Mobile Networks and Applications*, 10(4):563–579, 2005.

[34] Vijay Bhuse and Ajay Gupta. Anomaly intrusion detection in wireless sensor networks. *J. High Speed Netw.*, 15(1):33–51, 2006.

[35] John Bicket, Daniel Aguayo, Sanjit Biswas, and Robert Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 31–42, New York, NY, USA, 2005. ACM.

[36] Bluetooth SIG. http://www.bluetooth.org. [Online; accessed 14.05.2009].

[37] Jeremy Blum, Min Ding, Andrew Thaeler, and Xiuzhen Cheng. *Handbook of Combinatorial Optimization*. Springer US, 2006.

[38] Bodymedia sensewear. http://www.sensewear.com/solutions_bms.php. [Online; accessed 14.05.2009].

[39] Cristian Borcea, Chalermek Intanagonwiwat, Porlin Kang, Ulrich Kremer, and Liviu Iftode. Spatial programming using smart messages: Design and implementation. In *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)*, pages 690–699, Washington, DC, USA, 2004. IEEE Computer Society.

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

[41] A. Boulis. Castalia: revealing pitfalls in designing distributed algorithms in wsn. In *Proceedings of the 5th international conference on Embedded networked sensor systems (Demo session)*, Sydney, Australia, November 2007.

[42] Athanassios Boulis, Chih-Chieh Han, and Mani B. Srivastava. Design and implementation of a framework for efficient and programmable sensor networks. In *Proc. of the 1st Intl. Conf. on Mobile Systems, Applications, and Services (MobiSys 2003)*, 2003.

[43] David Braginsky and Deborah Estrin. Rumor routing algorthim for sensor networks. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 22–31, New York, NY, USA, 2002. ACM.

[44] Scott Brave and Clifford Nass. Emotion in human-computer interaction. In Julie A. Jacko and Andrew Sears, editors, *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*, pages 81–96. L. Erlbaum Associates Inc., 2003.

[45] Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, John Heidemann, Ahmed Helmy, Polly Huang, Steven McCanne, Kannan Varadhan, Ya Xu, and Haobo Yu. Advances in network simulation. *IEEE Computer*, 33(5):59–67, May 2000. Expanded version available as USC TR 99-702b at http://www.isi.edu/~johnh/PAPERS/Bajaj99a.html.

[46] BTnodes. a distributed environment for prototyping ad hoc networks. http://www.btnode.ethz.ch. [Online; accessed 14.05.2009].

[47] Sonja Buchegger and Jean-Yves Le Boudec. Performance analysis of the confidant protocol. In *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 226–236, New York, NY, USA, 2002. ACM.

[48] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low-cost outdoor localization for very small devices. *Personal Communications, IEEE [see also IEEE Wireless Communications]*, 7(5):28–34, Oct 2000.

[49] P. Buonadonna, D. Gay, J. M. Hellerstein, W. Hong, and S. Madden. Task: Sensor network in a box. In *EWSN 2005*, 2005.

[50] F. Burkhardt, M. Van Ballegooy, R. Englert, and R. Huber. An emotion-aware voice portal. proc. electronic speech signal processing essp. In *burkhardt*, 2005.

[51] TAC by Schneider Electric. Wireless controller networks for building automation. benefits and opportunities for facility owners. *White Paper*, 2006.

[52] M. Caccamo and L. Y. Zhang. An implicit prioritized access protocol for wireless sensor networks. In *Proceedings of the 23rd IEEE Real-Time Syst. Symp. (RTSS'02)*, pages 39–48, Austin, TX, USA, December 2002.

[53] Giuseppe Caire and K. Raj Kumar. Information Theoretic Foundations of Adaptive Coded Modulation. *Proceedings of the IEEE*, 95(12):2274–2298, December 2007.

[54] Giuseppe Caire, Giorgio Taricco, and Ezio Biglieri. Optimum power control over fading channels. *IEEE Transactions on Information Theory*, 45(5):1468–1489, July 1999.

[55] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, Sep. 2002.

[56] CAN in Automation. http://www.can-cia.org. [Online; accessed 14.05.2009].

[57] Guohong Cao, G. Kesidis, T. La Porta, Bin Yao, and Shashi Phoha. *Purposeful Mobility in Tactical Sensor Networks*, chapter 3.3. IEEE Press, 2005.

[58] Qing Cao, Tarek Abdelzaher, John Stankovic, and Tian He. The liteos operating system: Towards unix-like abstractions for wireless sensor networks. In *Proceedings of the 7th international conference on Information processing in sensor networks (IPSN 2008)*, pages 233–244, Washington, DC, USA, April 2008. IEEE Computer Society.

[59] Srdjan Capkun, Jean pierre Hubaux, and Levente Buttyan. Mobility helps peer-to-peer security. *IEEE Transactions on Mobile Computing*, 5:43–51, 2003.

[60] Casino Lab, Colorado School of Mines. http://toilers.mines.edu/Public/. [Online; accessed: 14.05.2009].

[61] A. Cerpa, N. Busek, and D. Estrin. SCALE: A tool for Simple Connectivity Assessment in Lossy Environments. *CENS, UCLA, Tech. Rep*, 2003.

[62] A. Cerpa, J.L. Wong, M. Potkonjak, and D. Estrin. Temporal properties of low power wireless links: modeling and implications on multi-hop routing. *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 414–425, 2005.

[63] Alberto Cerpa and Deborah Estrin. ASCENT: Adaptive Self-Configuring sEnsor Networks Topologies. *IEEE Transactions on Mobile Computing*, 3(3):272–285, September 2004.

[64] Alberto Cerpa, Jennifer L. Wong, Louane Kuang, Miodrag Potkonjak, and Deborah Estrin. Statistical model of lossy links in wireless sensor networks. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 11, Piscataway, NJ, USA, 2005. IEEE Press.

[65] Ugur Cetintemel, Andrew Flinders, and Ye Sun. Power-efficient data dissemination in wireless sensor networks. In *MobiDe '03: Proceedings of the 3rd ACM international workshop on Data engineering for wireless and mobile access*, pages 1–8, New York, NY, USA, 2003. ACM.

[66] Haowen Chan, Virgil D. Gligor, Adrian Perrig, and Gautam Muralidharan. On the distribution and revocation of cryptographic keys in sensor networks. *IEEE Trans. Dependable Secur. Comput.*, 2(3):233–247, 2005.

[67] S. Chan, C. Kaler, T. Kuehnel, A. Regnier, B. Roe, D. Sather, J. Schlimmer, H. Sekine, D. Walter, J. Weast, D. Whitehead, and D. Wright. Devices Profile for Web Services. Microsoft Developers Network Library, http://specs.xmlsoap.org/ws/2005/05/devprof/devicesprofile.pdf, May 2005. [Online; accessed 14.05.2009].

[68] Nicholas Chang and Mingyan Liu. Revisiting the ttl-based controlled flooding search: optimality and randomization. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 85–99, New York, NY, USA, 2004. ACM.

[69] Canfeng Chen and Jian Ma. Mobile enabled large scale wireless sensor networks. *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference*, 1:333–338, Feb. 2006.

[70] K. Chen, Y. Xue, S.H. Shah, and K. Nahrstedt. Understanding bandwidth-delay product in mobile ad hoc networks. *Computer Communications*, 27(10):923–934, June 2004.

[71] R. Chen, K. C. Chua, B. T. Tan, and C. S. Ng. Adaptive error coding using channel prediction. *Wireless Networks*, 5(1):23–32, February 1999.

[72] M. Chiang. Balancing transport and physical Layers in wireless multihop networks: jointly optimal congestion control and power control. *IEEE Journal on Selected Areas in Communications*, 23(1):104–116, 2005.

[73] Carla-Fabiana Chiasserini and Ramesh R. Rao. Coexistence mechanisms for interference mitigation in the 2.4-ghz ism band. *IEEE Transactions on Wireless Communications*, 2(5):964–975, September 2003.

[74] C-Y. Chong and S. P. Kumar. Sensor networks: Evolution, opportunities and challenges. *Proc. of the IEEE*, 91:1247–1256, 2003.

[75] D.R. Choudhary, Dev Anshul, S. Roy, and C.S. Thejaswi. Computationally and resource efficient group key agreement for ad hoc sensor networks. *Communication Systems Software and Middleware, 2007. COMSWARE 2007. 2nd International Conference on*, pages 1–10, Jan. 2007.

[76] Tanzeem Choudhury, Gaetano Borriello, Sunny Consolvo, Dirk Haehnel, Beverly Harrison, Bruce Hemingway, Jeffrey Hightower, Pedja Klasnja, Karl Koscher, Anthony LaMarca, Jonathan Lester, James Landay, Louis Legrand, Ali Rahimi, Adam Rea, and Danny Wyatt. The mobile sensing platform: An embedded activity recognition system. *IEEE Pervasive Computing*, 7(2), 2008.

[77] B.N. Chun, P. Buonadonna, A. AuYoung, Chaki Ng, D.C. Parkes, J. Shneidman, A.C. Snoeren, and A. Vahdat. Mirage: A microeconomic resource allocation system for sensornet testbeds. *Embedded Networked Sensors, 2005. EmNetS-II. The Second IEEE Workshop on*, pages 19–28, May 2005.

[78] A. Cilardo, L. Coppolino, N. Mazzocca, and L. Romano. Elliptic curve cryptography engineering. *Proceedings of the IEEE*, 94(2):395–406, Feb. 2006.

[79] Jeffrey Considine, Feifei Li, George Kollios, and John Byers. Approximate aggregation techniques for sensor databases. In *ICDE '04: Proceedings of the 20th International Conference on Data Engineering*, page 449, Washington, DC, USA, 2004. IEEE Computer Society.

[80] Continua Alliance. http://www.continuaalliance.org. [Online; accessed: 14.05.2009].

[81] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme. Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle. In *in IEEE International Conference on Robotics and Automation*, pages 3602–3609, 2004.

[82] Paolo Costa, Luca Mottola, Amy L. Murphy, and Gian Pietro Picco. Teenylime: transiently shared tuple space middleware for wireless sensor networks. In *Proceedings of the international workshop on Middleware for sensor networks (MidSens '06)*, pages 43–48, New York, NY, USA, 2006. ACM.

[83] Paul Couderc and Michel Banâtre. Ambient computing applications: an experience with the spread approach. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03)*, page 291, Washington, DC, USA, 2003. IEEE Computer Society.

[84] R. L. Cruz and Arvind V. Santhanam. Optimal routing, link scheduling and power control in multi-hop wireless networks. In *Proc. IEEE INFOCOM*, San Francisco, CA, March 2003.

[85] S. Cui, R. Madan, A. Goldsmith, and S. Lall. Joint routing, MAC, and link layer optimization in sensor networks with energy constraints. In *IEEE International Conference on Communications (ICC)*, Seoul, Korea, May 2005.

[86] Shuguang Cui, Andrea J. Goldsmith, and Ahmad Bahai. Energy-constrained modulation optimization. *IEEE Transactions on Wireless Communications*, 2004. to appear.

[87] Carlo Curino, Matteo Giani, Marco Giorgetta, Alessandro Giusti, Amy L. Murphy, and Gian Pietro Picco. TinyLIME: Bridging mobile and sensor networks through middleware. In *Proc. of the 3rd IEEE Int'l Conf. on Pervasive Computing and Communications (PerCom 2005)*, pages 61–72, 2005.

[88] Ana Paula R. da Silva, Marcelo H. T. Martins, Bruno P. S. Rocha, Antonio A. F. Loureiro, Linnyer B. Ruiz, and Hao Chi Wong. Decentralized intrusion detection in wireless sensor networks. In *Q2SWinet '05: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, pages 16–23, New York, NY, USA, 2005. ACM.

[89] Datalogisk Institut (DIKU), University of Copenhagen. Re-Mote Testbed Framework. http://code.google.com/p/remote-testbed/. [Online; accessed: 14.05.2009].

[90] P. De, A. Raniwala, S. Sharma, and T. Chiueh. Mint: a miniaturized network testbed for mobile wireless research. *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, 4:2731–2742 vol. 4, March 2005.

[91] D.S.J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. *Wireless Networks*, 11(4):419–434, 2005.

[92] École Polytechnique Fédérale de Lausanne. Wasal testbed. http://wasal.epfl.ch/. [Online; accessed: 14.05.2009].

[93] Aitor del Coso, Umberto Spagnolini, and Christian Ibars. Cooperative Distributed MIMO Channels in Wireless Sensor Networks. *IEEE Journal on Selected Areas in Communications*, 25(2):402–414, February 2007.

[94] M. Demirbas, A. Arora, and V. Mittal. Floc: A fast local clustering service for wireless sensor networks. In *Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS)*, 2004.

[95] I. Demirkol, C. Ersoy, and F. Alagoz. Mac protocols for wireless sensor networks: a survey. *Communications Magazine, IEEE*, 44(4):115–121, April 2006.

[96] Jing Deng, Ben Liang, and P.K. Varshney. Tuning the carrier sensing range of IEEE 802.11 MAC. *GLOBECOM '04: IEEE Global Telecommunications Conference*, 5:2987–2991, Nov.-3 Dec. 2004.

[97] Natasha Devroye, Patrick Mitran, and Vahid Tarokh. Achievable rates in cognitive radio channels. *IEEE Transactions on Information Theory*, 52(5):1813–1827, May 2006.

[98] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[99] Digital Smart Home Services. http://www.ember.net/press_release.html?id=48. [Online; accessed: 14.05.2009].

[100] G. Dini, M. Pelagatti, and I.M. Savino. An algorithm for reconnecting Wireless Sensor Network partitions. *IEEE Proc. of EWSN*, pages 253–267, 2008.

[101] Gianluca Dini and Ida Maria Savino. S2rp: a secure and scalable rekeying protocol for wireless sensor networks. *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, pages 457–466, Oct. 2006.

[102] John R. Douceur. The sybil attack. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 251–260, London, UK, 2002. Springer-Verlag.

[103] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern classification. *Wiley Interscience*, 2nd ed.

[104] Alexandra Duel-Hallen. Fading Channel Prediction for Mobile Radio Adaptive Transmission Systems. *Proceedings of the IEEE*, 95(12):2299–2313, December 2007.

[105] A. Dunkels, B. Grönvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Workshop on Embedded Networked Sensors*, Tampa, Florida, USA, November 2004.

[106] A. Dunkels, F. Österlind, and Z. He. An adaptive communication architecture for wireless sensor networks. In *Proc. Fifth ACM Conference on Networked Embedded Sensor Systems (SenSys 2007)*, Sydney, Australia, November 2007.

[107] Adam Dunkels, Fredrik Österlind, Nicolas Tsiftes, and Zhitao He. Software-based on-line energy estimation for sensor nodes. In *Proceedings of the 4th Workshop on Embedded Networked Sensors (Emnets)*, Cork, Ireland, June 2007.

[108] Adam Dunkels, Oliver Schmidt, and Thiemo Voigt. Using protothreads for sensor node programming. In *Proceedings of the REALWSN'05 Workshop on Real-World Wireless Sensor Networks*, 2005.

[109] Mathilde Durvy, Julien Abeillé, Patrick Wetterwald, Colin O'Flynn, Blake Leverett, Eric Gnoske, Michael Vidales, Geoff Mulligan, Nicolas Tsiftes, Niclas Finne, and

Adam Dunkels. Making sensor networks ipv6 ready. In *Proceedings of the Sixth ACM Conference on Networked Embedded Sensor Systems (ACM SenSys 2008), poster session*, Raleigh, North Carolina, USA, November 2008. Best poster award.

[110] Prabal K. Dutta, Jonathan W. Hui, David C. Chu, and David E. Culler. Securing the deluge network programming system. In *IPSN '06: Proceedings of the 5th international conference on Information processing in sensor networks*, pages 326–333, New York, NY, USA, 2006. ACM.

[111] Matthias Dyer, Jan Beutel, Thomas Kalt, Patrice Oehen, Lothar Thiele, Kevin Martin, and Philipp Blum. Deployment Support Network - A Toolkit for the Development of WSNs. In *Proceedings of the 4th European Conference on Wireless Sensor Networks (EWSN)*, pages 195–211, Delft, The Netherlands, 2007.

[112] Jean-Pierre Ebert and Adam Wolisz. Combined Tuning of RF Power and Medium Access Control for WLANs. *MONET - Mobile Networks and Applications*, 6(5):417–426, September 2000.

[113] eCos. Open source, royalty-free, real-time operating system. http://ecos.sourceware.org. [Online; accessed 14.05.2009].

[114] J. Mikael Eklund, Thomas Riisgaard Hansen, Jonathan Sprinkle, and Shankar Sastry. Information Technology for Assisted Living at Home: building a wireless infrastructure for assisted living. In *Proceedings of the 27th Annual International Conference of the IEEE Engineering In Medicine and Biology Society (EMBS)*, pages 3931–3934, September 2005.

[115] Elderly monitoring project. http://www.ember.com/press_release.html?id=182. [Online; accessed: 14.05.2009].

[116] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *Proceedings of 5th symposium on Operating systems design and implementation (OSDI'02)*, pages 147–163, Boston, MA, USA, December 2002.

[117] Mohamed Eltoweissy, M. Hossain Heydari, Linda Morales, and I. Hal Sudborough. Combinatorial optimization of group key management. *J. Netw. Syst. Manage.*, 12(1):33–50, 2004.

[118] Sinem Coleri Ergen and Pravin Varaiya. Energy efficient routing with delay guarantee for sensor networks. *Wirel. Netw.*, 13(5):679–690, 2007.

[119] J. Eriksson, A. Dunkels, N. Finne, F. Österlind, and T. Voigt. MSPSim – an extensible simulator for MSP430-equipped sensor boards. In *Proceedings of the European Conference on Wireless Sensor Networks (EWSN), Poster/Demo session*, Delft, The Netherlands, January 2007.

[120] J. Eriksson, F. Österlind, N. Finne, N. Tsiftes, A. Dunkels, T. Voigt, R. Sauter, and P. Marrón. COOJA/MSPSim: interoperability testing for wireless sensor networks. In *International Conference on Simulation Tools and Techniques (SIMUTools)*, Rom, Italy, March 2009.

[121] Laurent Eschenauer and Virgil D. Gligor. A key-management scheme for distributed sensor networks. In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47, New York, NY, USA, 2002. ACM.

[122] European Commission DG Information Society & Media. Monitoring and control: today's market, its evolution till 2020 and the impact of ict on these. http://www.decision.eu/smart/SMART_9Oct_v2.pdf, October 2008. Workshop presentation.

[123] Eurostat. http://ec.europa.eu/eurostat/. [Online; accessed: 14.05.2009].

[124] I. Maza F. Caballero, L. Merino and A. Ollero. A particle filtering method for Wireless Sensor Network localization with an aerial robot beacon. In *Proc. of Intl. Conf. on Robotics and Automation (ICRA2008)*, 2008.

[125] Facereader: Affective-cognitive state inference from facial video, affective computing group. http://affect.media.mit.edu/projects.php?id=2523. [Online; accessed: 14.05.2009].

[126] S. P. Fekete, A. Kröller, D. Pfisterer, S. Fischer, and C. Buschmann. Neighborhood-based topology recognition in sensor networks. In *Proc. of the 1st Int. Workshop on Algorithmic Aspects of Wireless Sensor Networks*, 2004.

[127] Roy T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.

[128] Niclas Finne, Joakim Eriksson, Adam Dunkels, and Thiemo Voigt. Experiences from two sensor network deployments self-monitoring and self-configuration keys to success. In *Proceedings of WWIC 2008*, May 2008.

[129] Frank H. P. Fitzek and Marcos D. Katz, editors. *Cooperation in Wireless Networks: Principles and Applications – Real Egoistic Behaviour is to Cooperate!* Springer, Netherlands, 2006.

[130] Frank H. P. Fitzek and Marcos D. Katz, editors. *Cognitive Wireless Networks – Concepts, Methodologies and Visions Inspiring the Age of Enlightenment of Wireless Communications*. Springer, Netherlands, 2007.

[131] Elgar Fleisch and Friedemann Mattern, editors. *Das Internet der Dinge: Ubiquitous Computing und RFID in der Praxis:Visionen, Technologien, Anwendungen, Handlungsanleitungen*. Springer, 2005.

[132] Christian Frank and Kay Römer. Algorithms for generic role assignment in wireless sensor networks. In *Proceedings of the 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys)*, San Diego, CA, USA, November 2005.

[133] Stefan Funke. Topological hole detection in Wireless Sensor Networks and its applications. In *Proc. of the Joint Workshop on Foundations of Mobile Computing*, 2005.

[134] Stefan Funke and Christian Klein. Hole detection or: "how much geometry hides in connectivity?". In *Proc. of the 22nd Symp. on Computational Geometry*, 2006.

[135] Galvactivator project. http://www.media.mit.edu/galvactivator/index2.html. [Online; accessed: 14.05.2009].

[136] S. Ganeriwal, R. Kumar, and M. B. Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of 1st international conference on Embedded networked sensor systems (SenSys'03)*, pages 138–149, Los Angeles, California, USA, November 2003.

[137] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor networks. *UCLA Computer Science Technical Report UCLA/CSD-TR*, pages 02–0013, 2003.

[138] Deepak Ganesan, Ramesh Govindan, Scott Shenker, and Deborah Estrin. Highly-resilient, energy-efficient multipath routing in Wireless Sensor Networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(4):11–25, 2001.

[139] David Garlan, Dan Siewiorek, Asim Smailagic, and Peter Steenkiste. Project Aura: Toward Distraction-Free Pervasive Computing. *IEEE Pervasive Computing*, 1(2):22–31, 2002.

[140] Matthias Gauger, Pedro José Marrón, Daniel Kauker, and Kurt Rothermel. Low overhead assignment of symbolic coordinates in sensor networks. In *Proceedings of the IFIP International Conference on Wireless Sensor and Actor Networks (WSAN 2007)*, pages 179–190. Springer, September 2007.

[141] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler. The nesC language: A holistic approach to networked embedded systems. In *Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation*, pages 1–11, 2003.

[142] Gazebo - 3D multiple robot simulator with dynamics. http://playerstage.sourceforge.net/index.php?src=gazebo. [Online; accessed: 14.05.2009].

[143] Johannes Gehrke and Samuel Madden. Query processing in sensor networks. *Pervasive Computing*, 3(1):46–55, 2004.

[144] O. Gelbart, E. Leontie, B. Narahari, and R. Simha. Architectural support for securing application data in embedded systems. *Electro/Information Technology, 2008. EIT 2008. IEEE International Conference on*, pages 19–24, May 2008.

[145] Brian P. Gerkey, Richard T. Vaughan, and Andrew Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proc. IEEE Intl. Conf. on Advanced Robotics (ICAR2003)*, pages 317–323, Coimbra, Portugal, June 2003.

[146] S. Gezici, Zhi Tian, G.B. Giannakis, H. Kobayashi, A.F. Molisch, H.V. Poor, and Z. Sahinoglu. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *Signal Processing Magazine, IEEE*, 22(4):70–84, July 2005.

[147] Kajaldeep Ghumman. Location-aware combinatorial key management scheme for clustered sensor networks. *IEEE Trans. Parallel Distrib. Syst.*, 17(8):865–882, 2006. Senior Member-Mohamed F. Younis and Senior Member-Mohamed Eltoweissy.

[148] P.B. Gibbons, B. Karp, Y. Ke, S. Nath, and Srinivasan Seshan. Irisnet: an architecture for a worldwide sensor web. *Pervasive Computing, IEEE*, 2(4):22–33, Oct.-Dec. 2003.

[149] J. Gibson, G. Xie, and Y. Xiao. Performance limits of fair-access in sensor networks with linear and selected grid topologies. In *GLOBECOM '07: 50th IEEE Global Communications Conference Ad Hoc and Sensor Networking Symposium*, 2007.

[150] Lewis Girod, Thanos Stathopoulos, Nithya Ramanathan, Jeremy Elson, Deborah Estrin, Eric Osterweil, and Tom Schoellhammer. A system for simulation, emulation, and deployment of heterogeneous sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 201–213, New York, NY, USA, 2004. ACM Press.

[151] Robert Bosch GmbH, Stuttgart. CAN specification, ver. 2.0. http://www.semiconductors.bosch.de/pdf/can2spec.pdf, 1991. [Online; accessed: 14.05.2009].

[152] Omprakash Gnawali, Ki-Young Jang, Jeongyeup Paek, Marcos Vieira, Ramesh Govindan, Ben Greenstein, August Joki, Deborah Estrin, and Eddie Kohler. The tenet architecture for tiered sensor networks. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 153–166, New York, NY, USA, 2006. ACM.

[153] Guy Gogniat, Tilman Wolf, and Wayne Burleson. Reconfigurable security support for embedded systems. In *HICSS '06: Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, page 250.1, Washington, DC, USA, 2006. IEEE Computer Society.

[154] Philippe Golle, Dan Greene, and Jessica Staddon. Detecting and correcting malicious data in vanets. In *VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 29–37, New York, NY, USA, 2004. ACM.

[155] R. Gravina, A. Guerrieri, G. Fortino, F. Bellifemine, R. Giannantonio, and M. Sgroi. Development of body sensor network applications using spine. In *System, Man and Cybernetics Conference (SMC), Singapore*, 2008.

[156] ARC Advisory Group. Wireless Technology in Process Manufacturing. Worldwide Outlook - Market Analysis and Forecast Through 2010, 2006.

[157] ARC Advisory Group. Below the Wireless Infrastructure: Wireless Device Strategies for Discrete Manufacturers, 2008.

[158] E. Guenterberg, A. Yang, R. Jafari, R. Bajcsy, and S. Sastry. A lightweight and real-time fine-grained signal annotation technique based on markov models in body sensor networks. *IEEE Transactions on Information Technology in Biomedicine*, 2008.

[159] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, 1998.

[160] Ramakrishna Gummadi, Omprakash Gnawali, and Ramesh Govindan. Macro-programming Wireless Sensor Networks using Kairos. In *Proc. of the International Conference on Distributed Computing in Sensor Systems*, pages 126–140, 2005.

[161] M. Gunes, B. Blywis, and J. Schiller. A hybrid testbed for long-term Wireless Sensor Network studies. In *Int. Workshop on Sensor Network Engineering (IWSNE)*, 2008.

[162] G. Gupta and M. Younis. Fault-tolerant clustering of wireless sensor networks. In *WCNC '03: IEEE Wireless Communication and Networks Conference*. IEEEPress, 2003.

[163] Andreas Haag, Silke Goronzy, Peter Schaich, and Jason Williams. Emotion Recognition Using Bio-Sensors: First Steps Towards an Automatic System. In *Affective Dialogue Systems, Tutorial and Research Workshop (ADS 2004)*, pages 36–48, 2004.

[164] Z. J. Haas and J. Deng. Dual busy tone multiple access (dbtma)–a multiple access control scheme for ad hoc networks. *IEEE Transactions on Communications*, 50(6):975–985, 2002.

[165] Bo Han and Weijia Jia. Clustering wireless ad hoc networks with weakly connected dominating set. *J. Parallel Distrib. Comput.*, 67(6):727–737, 2007.

[166] C. Han, R. K. Rengaswamy, R. Shea, E. Kohler, and M. Srivastava. SOS: A dynamic operating system for sensor networks. In *MobiSys '05*, 2005.

[167] Qi Han, Iosif Lazaridis, Sharad Mehrotra, and Nalini Venkatasubramanian. Sensor data collection with expected reliability guarantees. In *PERCOMW '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 374–378, Washington, DC, USA, 2005. IEEE Computer Society.

[168] Zhu Han and K. J. Ray Liu. *Resource Allocation for Wireless Networks: Basics, Techniques, and Applications.* Cambridge University Press, 2008.

[169] Vlado Handziski, Andreas Köpke, Andreas Willig, and Adam Wolisz. Twist: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks. In *REALMAN '06: Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, pages 63–70, New York, NY, USA, 2006. ACM.

[170] Shinsuke Hara, Akira Ogino, Makoto Araki, Minoru Okada, and Norihiko Morinaga. Throughput Performance of SAW-ARQ Protocol with Adaptive Packet Length in Mobile Packet Data Transmission. *IEEE Transactions on Vehicular Technology*, 45(3):561–569, August 1996.

[171] I. Haratcherev, G. Halkes, T. Parker, O. Visser, and K. Langendoen. PowerBench: A Scalable Testbed Infrastructure for Benchmarking Power Consumption. In *Int. Workshop on Sensor Network Engineering (IWSNE)*, 2008.

[172] J. K. Hart and K. Martinez. Environmental sensor networks: A revolution in the earth system sience? *Earth-Science Reviews*, 11, 2006.

[173] Aamir Hasan, K. Yang, and J.G. Andrews. Clustered cdma ad hoc networks without closed-loop power control. *Military Communications Conference, 2003. MILCOM 2003. IEEE*, 2:1030–1035 Vol.2, Oct. 2003.

[174] Tilemahos Hasiotis, George Alyfantis, Vassileios Tstsos, Odysseas Sekkas, and Stathes Hadjiefthymiades. Sensation: A middleware integration platform for pervasive applications in wireless sensor networks. In *Proc. of the 2nd European Workshop on Wireless Sensor Networks*, pages 366–377, 2005.

[175] A. Hasler, I. Talzi, J. Beutel, C. Tschudin, and S. Gruber. Wireless sensor networks in permaforst research: Concepts, requirements, implementation, and challenges. In *NICOP 2008*, 2008.

[176] Mareca Hatler, Darryl Gurganious, Charlie Chi, and Mike Ritter. WSN for Smart Industries. OnWorld Study, 2007.

[177] Jan-Hinrich Hauer, Vlado Handziski, and Adam Wolisz. Experimental study of the impact of wlan interference on ieee 802.15.4 body area networks. In *Proc. 6th European Conference on Wireless Sensor Networks (EWSN)*, Cork, Ireland, February 2009.

[178] Simon Haykin. Cognitive Radio: Brain-Empowered Wireless Communications. *IEEE Journal on Selected Areas in Communications*, 23(2):201–220, February 2005.

[179] R. Hazlett and J. Benedek. Measuring emotional valence to understand the user's experience of software. *Int. J. Hum.-Comput. Stud.*, 65 (4)(pp 306-314), 2007.

[180] T. He, S. Krishnamurthy, J. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, G. Zhou, J. Hui, and B. Krogh. VigilNet: An integrated sensor network system for energy-efficient surveillance. *ACM Transactions on Sensor Networks*, 2(1):1–38, February 2006.

[181] Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, and Tarek Abdelzaher. Range-free localization schemes for large scale sensor networks. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 81–95, New York, NY, USA, 2003. ACM.

[182] Wendi B. Heinzelman, Amy L. Murphy, Hervaldo S. Carvalho, and Mark A. Perillo. Middleware to support sensor network applications. *IEEE Network*, 18(1):6–14, 2004.

[183] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless sensor networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS)*, pages 8020–8029, Hawaii, USA, January 2000.

[184] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 174–185, New York, NY, USA, 1999. ACM Press.

[185] Tristan Henderson and David Kotz. Measuring wireless LANs. In Rajeev Shorey, A. Ananda, Mun Choon Chan, and Wei Tsang Ooi, editors, *Mobile, Wireless, and Sensor Networks*, pages 5–29. Wiley, 2006.

[186] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for networked sensors. In *Proc. of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 93–104, November 2000.

[187] Thanongsak Himsoon, W. Pam Siriwongpairat, Zhu Han, and K. J. Ray Liu. Lifetime Maximization via Cooperative Nodes and Relay Deployment in Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 25(2):306–317, February 2007.

[188] L. Van Hoesel, T. Nieberg, J. Wu, and P. Havinga. Prolonging the Lifetime of Wireless Sensor Networks by Cross-Layer Interaction. *IEEE Wireless Communications*, 11(6):78–86, 2004.

[189] Fritz Hohl, Uwe Kubach, Alexander Leonhardi, Kurt Rothermel, and Markus Schwehm. Next century challenges: Nexus—an open global infrastructure for spatial-aware applications. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 249–255, New York, NY, USA, 1999. ACM.

[190] Tim Holliday and Andrea Goldsmith. Optimal power control and source-channel coding for delay constrained traffic over wireless channels. In *Proc. IEEE ICC 2002*, New York, April 2002. IEEE.

[191] Home automation using WSN. http://asterix.ist.utl.pt/rmr/WSN/CaseStudies/WSN_Home_Enviroment/arquitecturas.html. [Online; accessed 14.05.2009].

[192] S. P. Hoogendoorn and P. H. L. Bovy. State-of-the-art of vehicular traffic flow modelling. *Journal of Systems and Control Engineering - Special Issue on Road Traffic Modelling and Control*, 215(4):283–304, 2001.

[193] Jennifer Hou, Qixin Wang, Linda Ball, Stanley Birge, Marco Caccamo, Chin-Fei Cheah, Eric Gilbert, Carl Gunter, Elsa Gunter, Chang-Gun Lee, Karrie Karahalios, Min-Young Nam, Narasimhan Nitya, Chaudhri Rohit, Lui Sha, Wook Shin, Yang Yu, and Zheng Zeng. Pas: A wireless-enabled, sensor-integrated personal assistance system for independent and assisted living. In *HCMDSS 2007*, 2007.

[194] Andrew Howard, Maja J Matarić, and Gaurav S Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02)*, pages 299–308, 2002.

[195] Ivan Howitt and Jose A. Gutierrez. IEEE 802.15.4 low rate - wireless personal area network coexistence issues. In *Proc. Wireless Communications and Networking Conference 2003 (WCNC 2003)*, pages 1481–1486, New Orleans, Louisiana, March 2003.

[196] Yih-Chun Hu, D.B. Johnson, and A. Perrig. Sead: secure efficient distance vector routing for mobile wireless ad hoc networks. *Mobile Computing Systems and Applications, 2002. Proceedings Fourth IEEE Workshop on*, pages 3–13, 2002.

[197] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne: a secure on-demand routing protocol for ad hoc networks. *Wirel. Netw.*, 11(1-2):21–38, 2005.

[198] Zhihua Hu and Baochun Li. Fundamental performance limits of wireless sensor networks. In *in Ad Hoc and Sensor*. Nova Science Publishers, 2004.

[199] Hsing-Jung Huang, Ting-Hao Chang, Shu-Yu Hu, and Polly Huang. Magnetic diffusion: disseminating mission-critical data for dynamic sensor networks. In *MSWiM '05: Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 134–141, New York, NY, USA, 2005. ACM.

[200] J. W. Hui and D. Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. In *Proc. SenSys'04*, Baltimore, Maryland, USA, November 2004.

[201] Humboldt Universität zu Berlin. Berlin Roof Net. http://www.berlinroofnet.de/. [Online; accessed: 14.05.2009].

[202] T. Huynh and B. Schiele. Analyzing features for activity recognition. In *2005 Joint conference on smart objects and ambient intelligence: innovative context-aware services: usage and technologies,2005*, 2005.

[203] L. Hwang. Grouping strategy for solving hidden node problem in IEEE 802.15.4 LR-WPAN. In *WICON'05: Proc. of the 1st International Conference on Wireless Internet*, pages 26–32. IEEEPress, 2005.

[204] Institute of Electrical and Electronics Engineers, Inc. *IEEE Standard 802 for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999.

[205] Institute of Electrical and Electronics Engineers, Inc. *IEEE Standard 11073 for Health informatics – Point-of-care medical device communication*, 2004.

[206] Institute of Electrical and Electronics Engineers, Inc. *IEEE Standard 802 for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)*, September 2006.

[207] Chalermek Intanagonwiwat, Rajesh Gupta, and Amin Vahdat. Declarative Resource Naming for Macroprogramming Wireless Networks of Embedded Systems. In *Proceedings of the Second International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2006)*, pages 192–199. Springer, Jul 2006.

[208] Mika Ishizuka and Masaki Aida. Performance study of node placement in sensor networks. In *ICDCSW '04: Proceedings of the 24th International Conference on Distributed Computing Systems Workshops - W7: EC (ICDCSW'04)*, pages 598–603, Washington, DC, USA, 2004. IEEE Computer Society.

[209] V. Isler, S. Kannan, and K. Daniilidis. Sampling based sensor-network deployment. *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, 2:1780–1785 vol.2, Sept.-2 Oct. 2004.

[210] Toshiki Iso, Norihiro Kawasaki, and Shoji Kurakake. Personal context extractor with multiple sensors on a cell phone. In *7th IFIP International conference on Mobile and Wireless Communication Networks (MWCN 2005)*, September 2005.

[211] J. Jeong, S. Kim, and A. Broad. Network reprogramming. TinyOS documentation, 2003. Visited 2006-04-06.

[212] Jaein Jeong and David Culler. Incremental network programming for wireless sensors. In *The First IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON 2004)*, 2004.

[213] Xiaofan Jiang, Jay Taneja, Jorge Ortiz, Arsalan Tavakoli, Prabal Dutta, Jaein Jeong, David Culler, Philip Levis, Scott Shenker, and Arch Rock Corporation. An architecture for energy management in wireless sensor networks. In *In International Workshop on Wireless Sensor Network Architecture (WSNA07*, page 2007. ACM Press, 2007.

[214] D. Johnson, T. Stack, R. Fish, D. M. Flickinger, L. Stoller, R. Ricci, and J. Lepreau. Mobile emulab: A robotic wireless and sensor network testbed. *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, April 2006.

[215] M. R. Jongerden and B. R. H. M. Haverkort. Battery modeling. Technical Report TR-CTIT-08-01, Centre for Telematics and Information Technology, University of Twente, Enschede, January 2008.

[216] Glenn Judd and Peter Steenkiste. Providing contextual information to pervasive computing applications. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom '03)*, page 133, Washington, DC, USA, 2003. IEEE Computer Society.

[217] P. Jurcik, R. Severino, A. Koubâa, M. Alves, and E. Tovar. Real-time communications over cluster-tree sensor networks with mobile sink behaviour. In *RTCSA '08: Proc. of the 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, 2008.

[218] Samir Kallel. Efficient hybrid arq protocols with adaptive forward error correction. *IEEE Transactions on Communications*, 42(2):281–289, February 1994.

[219] S. Kallio, J. Kela, and J. Mantyjarvi. Online gesture recognition system for mobile interaction. In *IEEE International Conference on Systems, Man and Cybernetics*, 2003.

[220] A. Kansal, S. Nath, J. Liu, and F. Zhao. Senseweb: An infrastructure for shared sensing. *IEEE MultiMedia*, 14(4), 2007.

[221] Tufan C. Karalar and Jan Rabaey. An rf tof based ranging implementation for sensor networks. In *Proceedings of the IEEE International Conference on Communications (ICC 2006)*, 2006.

[222] Chris Karlof, Naveen Sastry, and David Wagner. Tinysec: a link layer security architecture for wireless sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 162–175, New York, NY, USA, 2004. ACM Press.

[223] Chris Karlof and David Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *In First IEEE International Workshop on Sensor Network Protocols and Applications*, pages 113–127, 2003.

[224] P. Karn. MACA - a new channel access method for packet radio. In *Proceedings of the ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, 1990.

[225] S. Karnouskos, O. Baecker, L. M. S. de Souza, and P. Spiess. Integration of soa-ready networked embedded devices in enterprise systems via a cross-layered web service infrastructure. In *Proc. ETFA Emerging Technologies & Factory Automation IEEE Conference on*, pages 293–300, 25–28 Sept. 2007.

[226] Stamatis Karnouskos and Mian Mohammad Junaid Tariq. An agent-based simulation of soa-ready devices. In *Proc. Tenth International Conference on Computer Modeling and Simulation UKSIM 2008*, pages 330–335, 1–3 April 2008.

[227] Stamatis Karnouskos and Mian Mohammad Junaid Tariq. Using multi-agent systems to simulate dynamic infrastructures populated with large numbers of web service enabled devices. In *The 9th International Symposium on Autonomous Decentralized Systems (ISADS 2009), Athens, Greece, March 23-25, 2009*, 2009.

[228] Stamatis Karnouskos and Orestis Terzidis. Towards an information infrastructure for the future internet of energy. In *Kommunikation in Verteilten Systemen (KiVS 2007) Conference*. VDE Verlag, 26 Feb 2007 - 02 Mar 2007 2007.

[229] Stamatis Karnouskos, Orestis Terzidis, and Panagiotis Karnouskos. An advanced metering infrastructure for future energy networks. In *IFIP/IEEE 1st International Conference on New Technologies, Mobility and Security (NTMS 2007)*, pages 597–606. Springer, 2007.

[230] Vikas Kawadia and P. R. Kumar. A Cautionary Perspective on Cross-Layer Design. *IEEE Wireless Communications*, 12(1):3–11, February 2005.

[231] Vikas Kawadia and P.R. Kumar. Power control and clustering in ad hoc networks. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, 1:459–469 vol.1, March-3 April 2003.

[232] Mohammad Maifi Hasan Khan, Liqian Luo, Chengdu Huang, and Tarek Abdelzaher. SNTS: Sensor network troubleshooting suite. In *Proceedings of the 3rd International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 142–157, Santa Fe, New Mexico, USA, June 2007.

[233] J. Kim, E. Kim, S. Kim, D. Kim, and W. Lee. Low-energy localized clustering: An adaptive cluster radius configuration scheme for topology control in wireless sensor networks. In *Vehicular Technology Conference, 2005*, 2005.

[234] Network of Readers, Kimaldi Website. http://www.kimaldi.com/kimaldi_eng/productos/lectores_de_tarjetas/red_de_lectores_can/red_de_lectores_ampliacion_de_informacion. [Online; accessed: 14.05.2009].

[235] Paul Kocher, Joshua Ja E, and Benjamin Jun. Differential power analysis. In *Lecture Notes in Computer Science*, volume 1666, pages 388–397. Springer-Verlag, 1999.

[236] Nathan Koenig and Andrew Howard. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robot Systems (IROS 2004)*, pages 2149–2154, 2004.

[237] A. Köpke, V. Handziski, J.-H. Hauer, and H. Karl. Structuring the information flow in component-based protocol implementations for wireless sensor nodes. In *Proc. of Work-in-Progress Session of the 1st European Workshop on Wireless Sensor Networks (EWSN)*, Berlin, Germany, January 2004.

[238] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P.T. Klein Haneveld, T.E.V. Parker, O.W. Visser, H.S. Lichte, and S. Valentin. Simulating wireless and mobile networks in OMNeT++: The MiXiM vision. In *First International OMNeT++ Developers Workshop*, Marseille, France, March 2008.

[239] Joel Koshy and Raju Pandey. Remote incremental linking for energy-efficient reprogramming of sensor networks. In *Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN 2005)*, 2005.

[240] Joel Koshy and Raju Pandey. VM*: Synthesizing scalable runtime environments for sensor networks. In *Proc. of the 3rd Int'l. Conf. on Embedded Networked Sensor Systems*, pages 243–254, 2005.

[241] Venkata A. Kottapalli, Anne S. Kiremidjian, Jerome P. Lynch, Ed Carryer, Thomas W. Kenny, Kincho H. Law, and Ying Lei. Two-tiered Wireless Sensor Network architecture for structural health monitoring. In *SPIE '03: Proc. of 10th Annual International Symposium on Smart Structures and Materials*, pages 8–19, 2003.

[242] D. Kotz, C. Newport, and C. Elliott. The mistaken axioms of wireless-network research. *Computer Science, Dartmouth College, Technical Report TR2003-467, July*, 2003.

[243] A. Koubâa, M. Alves, and E. Tovar. A comprehensive simulation study of slotted csma/ca for ieee 802.15.4 wireless sensor networks. In *Proceedings of 5th IEEE International Workshop on Factory Communication Systems (WFCS'06)*, pages 183–192, Torino, Italy, June 2006.

[244] A. Koubâa, M. Alves, and E. Tovar. GTS allocation analysis in IEEE 802.15.4 for real-time wireless sensor networks. In *Proceedings of 14th International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS'06)*, Rhodes Island, Greece, April 2006.

[245] A. Koubâa, M. Alves, and E. Tovar. i-GAME: An implicit gts allocation mechanism in ieee 802.15.4. In *Proceedings of 18th Euromicro Conference on Real-Time Systems (ECRTS'06)*, Dresden, Germany, July 2006.

[246] A. Koubâa, M. Alves, and E. Tovar. IEEE 802.15.4: a federating communication protocol for time-sensitive wireless sensor networks. Technical Report HURRAY-TR-060202, IPP-HURRAY! Research Group, Institute Polytechnic Porto, Porto, Portugal, 2006.

[247] A. Koubâa, M. Alves, and E. Tovar. Improving the IEEE 802.15.4 slotted CSMA/CA MAC for time-critical events in wireless sensor networks. In *Proceedings of 5th International Workshop on Real-Time Networks (RTN'06)*, Dresden, Germany, July 2006.

[248] A. Koubâa, M. Alves, and E. Tovar. Energy/delay trade-off of the GTS allocation mechanism in IEEE 802.15.4 for wireless sensor networks. *International Journal of Communication Systems*, 20(7):791–808, July 2007.

[249] A. Koubâa, M. Alves, E. Tovar, and A. Cunha. An implicit GTS allocation mechanism in IEEE 802.15.4: Theory and practice. *Real-Time Systems Journal*, 39:169–204, 2008.

[250] Anis Koubaa, Mario Alves, and Eduardo Tovar. Modeling and worst-case dimensioning of cluster-tree wireless sensor networks. In *RTSS'06: Proc. of the 27th IEEE Real-Time Systems Symposium*, pages 412–421. IEEEPress, 2006.

[251] Daniel Krajzewicz, Georg Hertkorn, Christian Rössel, and Peter Wagner. Sumo (simulation of urban mobility); an open-source traffic simulation. In *Proceedings of the 4th Middle East Symposium on Simulation and Modelling (MESM2002)*, pages 183–187, 2002.

[252] Matthias Kranz, Radu Bogdan Rusu, Alexis Maldonado, Michael Beetz, and Albrecht Schmidt. A Player/Stage System for Context-Aware Intelligent Environments. In *Proc. Systems Support for Ubiquitous Computing Workshop (UbiSys 2006), at the 8th Annual Conference on Ubiquitous Computing (Ubicomp 2006)*, Orange County, California, September 2006.

[253] Sita Srinivasaraghavan Krishnakumar. Intelligent actor mobility in wireless sensor and actor networks. *Telecommunication Systems*, 2008.

[254] B. Krishnamachari and J. Ahn. Optimizing Data Replication for Expanding Ring-based Queries in Wireless Sensor Networks. *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*, pages 1–10, 2006.

[255] A. Kröller, S. P. Fekete, D. Pfisterer, and S. Fischer. Deterministic boundary recognition and topology extraction for large sensor networks. In *Proc. of the 17th Symp. on Discrete Algorithms*, 2006.

[256] Marwan Krunz, Alaa Muqattash, and Sung-Ju Lee. Transmission power control in wireless ad hoc networks: Challenges, solutions, and open issues. *IEEE Network Magazine*, 18(5):8–14, September 2004.

[257] YoungMin Kwon and Gul Agha. Passive localization: Large size sensor network localization based on environmental events. In *IPSN '08: Proceedings of the 2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*, pages 3–14, Washington, DC, USA, 2008. IEEE Computer Society.

[258] A. Lachenmann, P. Marrón, D. Minder, M. Gauger, O. Saukh, and K. Rothermel. TinyXXL: Language and runtime support for cross-layer interactions. In *Proceedings of the Third Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 178–187, 2006.

[259] Andreas Lachenmann, Pedro José Marrón, Daniel Minder, Olga Saukh, Matthias Gauger, and Kurt Rothermel. Versatile support for efficient neighborhood data sharing. In *Proceedings of the Fourth European Conference on Wireless Sensor Networks (EWSN 2007)*, LNCS, pages 1–16, 2007.

[260] A. Lahiri, S. Agarwal, A. Basu, and B.B. Bhattacharya. Recovery-based real-time static scheduling for battery life optimization. *VLSI Design, 2006. Held jointly with 5th International Conference on Embedded Systems and Design., 19th International Conference on*, pages 4 pp.–, Jan. 2006.

[261] O. Landsiedel, K. Wehrle, and S. Gotz. Accurate prediction of power consumption in sensor networks. In *EmNets '05: Proceedings of the 2nd IEEE workshop on Embedded Networked Sensors*, pages 37–44, Washington, DC, USA, 2005. IEEE Computer Society.

[262] Koen Langendoen and Niels Reijers. Distributed Localization in Wireless Sensor Networks: A Quantitative Comparison. *Computer Networks*, 43(4):499–518, 2003.

[263] J. Leal, A. Cunha, M. Alves, and A. Koubaa. On a IEEE 802.15.4/ZigBee to IEEE 802.11 gateway for the ART-WiSe architecture. In *ETFA '07: Work-in-Progress session of the 12th IEEE Conference on Emerging Technologies and Factory Automation*. IEEEPress, 2007.

[264] H. Lee, A. Cerpa, and P. Levis. Improving wireless simulation through noise modelling. In *IPSN '07*, 2007.

[265] M. Lehning, M. Bavay, H. Loewe, M. Parlange, and K. Aberer. Das swiss experiment und die zukunft der vorhersage von alpinen naturgefahren. In *Forum fuer Wissen 2007*, 2007.

[266] Paul Lettieri and Mani Srivastava. Adaptive frame length control for improving wireless link throughput, range and energy efficiency. In *Proc. INFOCOM 1998*, pages 564–571, San Francisco, CA, 1998. IEEE.

[267] P. Levis, N. Patel, D. Culler, and S. Shenker. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *Proc. NSDI'04*, March 2004.

[268] Philip Levis and David Culler. Matè: A tiny virtual machine for sensor networks. In *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*, pages 85–95, 2002.

[269] Philip Levis, David Gay, and David Culler. Active sensor networks. In *Proceedings of the 2nd USENIX/ACM Symposium on Network Systems Design and Implementation (NSDI)*, May 2005.

[270] Philip Levis, Nelson Lee, Matt Welsh, and David Culler. TOSSIM: Accurate and scalable simulation of entire tinyos applications. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 126–137, Los Angeles, California, USA, November 2003.

[271] Jinghao Li, Yuebin Bai, Haixing Ji, Jihong Ma, Yong Tian, and Depei Qian. Power: Planning and deployment platform for wireless sensor networks. *Grid and Cooperative Computing Workshops, 2006. GCCW '06. Fifth International Conference on*, pages 432–436, Oct. 2006.

[272] Shuoqi Li, Ying Lin, Sang Son, John Stankovic, and Yuan Wei. Event detection services using data service middleware in distributed sensor networks. In *Proceedings of the 2nd International Conference on Information Processing in Sensor Networks (IPSN'03)*, 2003.

[273] Jai-Jin Lim, D.L. Kiskis, and K.G. Shin. System support for management of networked low-power sensors. *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP*, pages 436–447, April 2006.

[274] C. L. Lisetti and F. Nasoz. Using non-invasive wearable computers to recognize human emotions from physiological signals. *EURASIP Journal on Applied Signal Processing - Special Issue on Multimedia Human-Computer Interface*, 11, 2004.

[275] Benyuan Liu, Peter Brass, Olivier Dousse, Philippe Nain, and Don Towsley. Mobility improves coverage of sensor networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 300–308, New York, NY, USA, 2005. ACM.

[276] Hongzhou Liu, Tom Roeder, Kevin Walsh, Rimon Barr, and Emin Gün Sirer. Design and implementation of a single system image operating system for ad hoc networks. In *MobiSys '05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 149–162, New York, NY, USA, 2005. ACM Press.

[277] Ting Liu and Margaret Martonosi. Impala: A middleware system for managing autonomic, parallel sensor systems. In *Proc. of the 9th ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming*, pages 107–118, 2003.

[278] C. Lombriser, D. Roggen, M. Stiger, and G. Trister. Titan: A tiny task network for dynamically reconfigurable heterogeneous sensor networks. *In: 15. Fachtagung Kommunikation in Verteilten Systemen (KiVS)*, 15(127-138), 2007.

[279] Chong Eik Loo, Mun Yong Ng, Christopher Leckie, and Marimuthu Palaniswami. Intrusion detection for routing attacks in sensor networks. *IJDSN*, 2(4):313–332, 2006.

[280] K. Lorincz, B. Chen, J. Waterman, G. Werner-Allen, and M. Welsh. Resource Aware Programming in the Pixie OS. In *ACM Conference on Networked Embedded Sensor Systems (SenSys 2008)*, Raleigh, NC, USA, November 2008.

[281] Konrad Lorincz, Bor rong Chen, Jason Waterman, Geoff Werner-Allen, and Matt Welsh. Pixie: An Operating System for Resource-Aware Programming of Embedded Sensors. In *Fifth Workshop on Embedded Networked Sensors (HotEmNets'08)*, June 2008.

[282] H. Lundgren, D. Lundberg, J. Nielsen, E. Nordstrom, and C. Tschudin. A large-scale testbed for reproducible ad hoc protocol evaluations. *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, 1:412–418 vol.1, Mar 2002.

[283] J. Luo, S. Blum, L. J. Cimini, L. J. Greenstein, and A. M. Haimovich. Decode-and-forward cooperative diversity with power allocation in wireless networks. *IEEE Transactions on Wireless Communications*, 6(3):793–799, March 2007.

[284] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. In *OSDI '02: Proceedings of the 5th symposium on Operating systems design and implementation*, pages 131–146, New York, NY, USA, 2002. ACM.

[285] Samuel R. Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. Tinydb: an acquisitional query processing system for sensor networks. *ACM Trans. Database Syst.*, 30(1):122–173, 2005.

[286] Sascha Mahlke, Michael Minge, and Manfred Thüring. Measuring multiple components of emotions in interactive contexts. In *Conference on Human Factors in Computing Systems (CHI '06), extended abstracts*, pages 1061–1066, 2006.

[287] Geoffrey Mainland, Greg Morrisett, and Matt Welsh. Flask: staged functional programming for sensor networks. In *Proceeding of the 13th ACM SIGPLAN international conference on Functional programming (ICFP '08)*, pages 335–346, New York, NY, USA, 2008. ACM.

[288] Geoffrey Mainland, David C. Parkes, and Matt Welsh. Decentralized, adaptive resource allocation for sensor networks. In *NSDI'05: Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation*, pages 315–328, Berkeley, CA, USA, 2005. USENIX Association.

[289] David Malan, Thaddeus Fulford-Jones, Matt Welsh, and Steve Moulton. Codeblue: An ad hoc sensor network infrastructure for emergency medical care. In *MobiSys 2004 Workshop on Applications of Mobile Embedded Systems (WAMES 2004)*, 2004.

[290] David J. Malan, Matt Welsh, and Michael D. Smith. Implementing public-key infrastructure for sensor networks. *ACM Trans. Sen. Netw.*, 4(4):1–23, 2008.

[291] R. Mangharam and R. Rajkumar. MAX: A maximal transmission concurrency mac for wireless networks with regular structure. In *Proceedings of IEEE Third International Conference on Broadband Communications, Networks and Systems (IEEE BROADNETS'06)*, pages 1–10, San Jose, CA, USA, October 2006.

[292] Raminder P. Mann, Kamesh R. Namuduri, and Ravi Pendse. Energy-aware routing protocol for ad hoc wireless sensor networks. *EURASIP J. Wirel. Commun. Netw.*, 5(5):635–644, 2005.

[293] J. Mantyjarvi, J. Himberg, and T. Seppanen. Recognizing human motion with multiple acceleration sensors. In *IEEE International Conference on Systems, Man and Cybernetics*, 2001.

[294] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi. The flooding time synchronization protocol. In *Proceedings of 2nd international conference on Embedded networked sensor systems (SenSys'04)*, pages 39–499, Baltimore, MD, USA, November 2004.

[295] Miklós Maróti, Péter Völgyesi, Sebestyén Dóra, Branislav Kusý, András Nádas, Ákos Lédeczi, György Balogh, and Károly Molnár. Radio interferometric geolocation. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 1–12, New York, NY, USA, 2005. ACM Press.

[296] P. Marron, D. Minder, and The Embedded WiSeNtsConsortium. Embedded wisents research roadmap. IST/FP6 (IST-004400), 2006.

[297] Pedro José Marrón, Matthias Gauger, Andreas Lachenmann, Daniel Minder, Olga Saukh, and Kurt Rothermel. Flexcup: A flexible and efficient code update mechanism for sensor networks. In *Proceedings of the Third European Workshop on Wireless Sensor Networks (EWSN 2006)*, pages 212–227, February 2006.

[298] Pedro José Marrón, Andreas Lachenmann, Daniel Minder, Jörg Hähner, Robert Sauter, and Kurt Rothermel. TinyCubus: A Flexible and Adaptive Framework for Sensor Networks. In Erdal Çayırcı Şebnem Baydere, and Paul Havinga, editors, *Proceedings of the 2nd European Workshop on Wireless Sensor Networks*, pages 278–289, Istanbul, Turkey, January 2005.

[299] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 255–265, New York, NY, USA, 2000. ACM.

[300] R.E. Mayagoitia, J.C.Lotters, P.H.Veltink, and H. Hermens. Standing balance evaluation using a triaxial accelerometer"sensing. *Gait and Posture*, 16(1), 2002.

[301] William P. McCartney and Nigamanth Sridhar. Abstractions for safe concurrent programming in networked embedded systems. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 167–180, New York, NY, USA, 2006. ACM.

[302] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M.B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 3:1380–1387 vol.3, 2001.

[303] Meraki Website. http://meraki.com. [Online: 14.05.2009].

[304] Chad Metcalf. TOSSIM live: Towards a testbed in a thread. Master's thesis, Colorado School of Mines, 2007.

[305] Steve Methley, Chris Davis, and John Parker. Transport final report. Technical report, Plextek Limited, Jun 2008. http://www.ofcom.org.uk/research/technology/research/sectorstudies/transport/.

[306] MetroSense. http://metrosense.cs.dartmouth.edu. [Online; accessed: 14.05.2009].

[307] Raquel A. F. Mini, Antonio A. F. Loureiro, and Badri Nath. The distinctive design characteristic of a wireless sensor network: the energy map. *Computer Communications*, 27(10):935 – 945, 2004. Protocol Engineering for Wired and Wireless Networks.

[308] A. Mishra, K. Nadkarni, and A. Patcha. Intrusion detection in wireless ad hoc networks. *Wireless Communications, IEEE*, 11(1):48–60, Feb 2004.

[309] MIT AgeLab. http://web.mit.edu/agelab/. [Online; accessed 14.05.2009].

[310] Joseph Mitola. The Software Radio Architecture. *IEEE Communications Magazine*, 33(5):26–38, May 1995.

[311] Joseph Mitola and Gerald Q. Maguire. Cognitive radio: Making software radios more personal. *IEEE Personal Communications*, 6(4):13–18, August 1999.

[312] M.J.Mathie, B.G.Celler, N.H.Lovell, and A.C.F.Coster. Classification of basic daily movements using a triaxial accelerometer. *Medical and Biological Engineering and Computing*, 42(5):679–687, September 2004.

[313] Eytan Modiano. An adaptive algorithm for optimizing the packet size used in wireless ARQ protocols. *Wireless Networks*, 5:279–286, 1999.

[314] R. Moe-Nilssen. A new method for evaluating motor control in gait under real-life environmental conditions. part 1. the instrument, part 2. gait analysis. *Clinical Biomechanics*, 13(pp 320-327), 1998.

[315] A. K. Mok and S. Ward. Distributed broadcast channel access. *Computer Networks*, 3(5):327–335, 1979.

[316] Ricardo Morla and Nigel Davies. Evaluating a location-based application: A hybrid test and simulation environment. In *In Proceedings of 2nd International Conference on Pervasive Computing*, 2004.

[317] Robert Morris, Eddie Kohler, John Jannotti, and M. Frans Kaashoek. The click modular router. *SIGOPS Oper. Syst. Rev.*, 33(5):217–231, 1999.

[318] Motescope Testbed, University of California, Berkeley. https://www.millennium.berkeley.edu/sensornets. [Online; accessed: 14.05.2009].

[319] Luca Mottola and Gian Pietro Picco. Logical neighborhoods: A programming abstraction for wireless sensor networks. In *Proc. of the Int'l Conf. on Distributed Computing in Sensor Systems*, pages 150–168, 2006.

[320] S. M. Mousavi, H. R. Rabiee, M. Moshref, and A. Dabirmoghaddam. Mobisim: A framework for simulation of mobility models in mobile ad-hoc networks. In *Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, 2007.

[321] Ali Muneeb, Voigt Thiemo, and Uzmi Zartash Afzal. Mobility management in sensor networks. In *Proceedings of Workshops Proceeding of 2nd IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS'06)*, page 10, San Francisco, California, 2006. Pages 131-140.

[322] Mirco Musolesi and Cecilia Mascolo. Mobility models for systems evaluation. *State of the Art on Middleware for Network Eccentric and Mobile Applications (MINEMA)*, 2008.

[323] K. Nahrstedt, S.H. Shah, and K. Chen. *Cross-Layer Architectures for Bandwidth Management in Wireless Networks*. Springer Verlag, Berlin, 2005.

[324] B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, and C.J. Bula. Ambulatory system for human motion analysis using a kinematic sensor: Monitoring of daily physical aactivity in the elderly. *IEEE Transactions on Biomedical Engineering*, 50(pp 771-723), 2003.

[325] B. Narendran, J. Sienicki, S. Yajnik, and P. Agrawal. Evaluation of an adaptive power and error control algorithm for wireless systems. In *Proc. Intl. Conf. on Communication (ICC)*, Montreal, Canada, June 1997.

[326] Asis Nasipuri and Kai Li. A directionality based location discovery scheme for wireless sensor networks. In *WSNA '02: Proceedings of the 1st ACM international workshop*

on Wireless sensor networks and applications, pages 105–111, New York, NY, USA, 2002. ACM.

[327] William Navidi and Tracy Camp. Stationary distributions for the random waypoint mobility model. *IEEE Transactions on Mobile Computing*, 3(1):99–108, Jan-Feb 2004.

[328] William Navidi, Tracy Camp, and Nick Bauer. Improving the Accuracy of Random Waypoint Simulations through Steady-State Initialization. In *Proceedings of the 15th International Conference on Modeling and Simulation*, pages 319–326, 2004.

[329] NetEquality Portland Deployment. http://www.netequality.org/map.php?id=hacienda. [Online; accessed: 14.05.2009].

[330] B.C. Neuman and T. Ts'o. Kerberos: an authentication service for computer networks. *Communications Magazine, IEEE*, 32(9):33–38, Sep 1994.

[331] Ryan Newton, Greg Morrisett, and Matt Welsh. The regiment macroprogramming system. In *Proceedings of the 6th international conference on Information processing in sensor networks (IPSN '07)*, pages 489–498, New York, NY, USA, 2007. ACM.

[332] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 151–162, New York, NY, USA, 1999. ACM.

[333] Yang Ni, Ulrich Kremer, Adrian Stere, and Liviu Iftode. Programming ad-hoc networks of mobile and resource-constrained devices. *SIGPLAN Not.*, 40(6):249–260, 2005.

[334] Dragoş Niculescu. Interference map for 802.11 networks. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 339–350, New York, NY, USA, 2007. ACM.

[335] Dragoş Niculescu and Badri Nath. Ad hoc positioning system (aps) using aoa. *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, 3:1734–1743, March 2003.

[336] Christopher Nitta, Raju Pandey, and Yann Ramin. Y-Threads: Supporting Concurrency in Wireless Sensor Networks. In *Proceedings of the Second IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS 2006)*, pages 169–184. Springer, Jun 2006.

[337] Nokia Step Counter. http://betalabs.nokia.com/betas/view/nokia-step-counter. [Online; accessed: 14.05.2009].

[338] Aria Nosratinia, Todd E. Hunter, and Ahmadreza Hedayat. Cooperative Communication in Wireless Networks. *IEEE Communications Magazine*, 42(10):74–80, October 2004.

[339] Omega Testbed, University of California, Berkeley. http://omega.cs.berkeley.edu/. [Online; accessed: 14.05.2009].

[340] I. Onat and A. Miri. An intrusion detection system for wireless sensor networks. *Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), IEEE International Conference on*, 3:253–259 Vol. 3, Aug. 2005.

[341] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-level sensor network simulation with cooja. In *Proceedings of the First IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2006)*, Tampa, Florida, USA, November 2006.

[342] M. Ott, I. Seskar, R. Siraccusa, and M. Singh. Orbit testbed software architecture: supporting experiments as a service. *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2005. Tridentcom 2005. First International Conference on*, pages 136–145, Feb. 2005.

[343] J. Padhye, S. Agarwal, V. Padmanabhan, L. Qiu, A. Rao, and B. Zill. Estimation of Link Interference in Static Multi-hop Wireless Networks. In *Proc. IMC 2005*, 2005.

[344] Justin Mazzola Paluska, Hubert Pham, Umar Saif, Grace Chau, Chris Terman, and Steve Ward. Structured decomposition of adaptive applications. In *Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '08)*, pages 1–10, Washington, DC, USA, 2008. IEEE Computer Society.

[345] Lilia Paradis and Qi Han. A survey of fault management in wireless sensor networks. *J. Netw. Syst. Manage.*, 15(2):171–190, 2007.

[346] Seung-Jong Park, Ramanuja Vedantham, Raghupathy Sivakumar, and Ian F. Akyildiz. A scalable approach for reliable downstream data delivery in wireless sensor networks. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 78–89, New York, NY, USA, 2004. ACM.

[347] Taejoon Park and K. G. Shin. Soft tamper-proofing via program integrity verification in wireless sensor networks. *Mobile Computing, IEEE Transactions on*, 4(3):297–309, 2005.

[348] Tamara Pazynyuk, JianZhong Li, George S. Oreku, and LiQiang Pan. Qos as means of providing wsns security. In *ICN '08: Proceedings of the Seventh International Conference on Networking (icn 2008)*, pages 66–71, Washington, DC, USA, 2008. IEEE Computer Society.

[349] N. Pereira, B. Andersson, and E. Tovar. Widom: A dominance protocol for wireless medium access. *IEEE Transactions on Industrial Informatics*, 3(2):120–130, May 2007.

[350] N. Pereira, A. Rowe, B. Andersson, and E. Tovar. Static-priority scheduling over wireless networks with multiple broadcast domains. In *RTSS'07: Proc. of the 28th IEEE International Real-Time Systems Symposium*, pages 447–458. IEEEPress, 2007.

[351] C. Perkins, E.M. Royer, and S.R. Das. Ad hoc On-Demand Distance Vector (AODV) Routing, 2003.

[352] J. K. Perng, B.D. Fisher, S. Hollar, and K.S.J. Pister. Acceleration sensing glove. In *ISWC*, 1999.

[353] Adrian Perrig, John Stankovic, and David Wagner. Security in wireless sensor networks. *Commun. ACM*, 47(6):53–57, 2004.

[354] Gian Pietro Picco, Amy L. Murphy, and Gruia-Catalin Roman. Lime: Linda meets mobility. In *Proceedings of the 21st international conference on Software engineering (ICSE '99)*, pages 368–377, New York, NY, USA, 1999. ACM.

[355] The Player Project - Free Software tools for robot and sensor applications. http://playerstage.sourceforge.net. [Online; accessed: 14.05.2009].

[356] Polar Sport and Fitness Monitor. http://www.polar.fi. [Online; accessed: 14.05.2009].

[357] J. Polley, D. Blazakis, J. McGee, D. Rusk, and JS Baras. ATEMU: a fine-grained sensor network simulator. In *The First IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, October 2004.

[358] Shankar R. Ponnekanti, Brad Johanson, Emre Kiciman, and Armando Fox. Portability, extensibility and robustness in iros. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom '03)*, page 11, Washington, DC, USA, 2003. IEEE Computer Society.

[359] K. Shashi Prabh and Tarek Abdelzaher. On scheduling and real-time capacity of hexagonal wireless sensor networks. In *ECRTS '07: Proc. of the 19th Euromicro Conference on Real-Time Systems*, pages 136–145. IEEE Press, Los Alamitos, CA, 2007.

[360] QNX RTOS. http://www.qnx.com. [Online; accessed: 14.05.2009].

[361] P. Rainville, A. Bechara, N. Naqvi, and A. Damasio. Basic emotions and associated with distinct patterns of cardiorespiratory activity. *International journal of psychophysiology : official journal of the International Organization of Psychophysiology*, 61(pp 5-18), 2006.

[362] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. In *Proceedings of 1st international conference on Embedded networked sensor systems (SenSys'03)*, pages 181–192, Los Angeles, California, USA, November 2003.

[363] Venkatesh Rajendran, Katia Obraczka, Yunjung Yi, Sung-Ju Lee, Ken Tang, and Mario Gerla. Combining source- and localized recovery to achieve reliable multicast in multi-hop ad hoc networks. In *NETWORKING 2004, Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communication, Third International IFIP-TC6 Networking Conference, Athens, Greece, May 9-14, 2004, Proceedings*, volume 3042 of *Lecture Notes in Computer Science*. Springer, 2004.

[364] Daler Rakhmatov, Sarma Vrudhula, and Deborah A. Wallach. Battery lifetime prediction for energy-aware computing. In *ISLPED '02: Proceedings of the 2002 international symposium on Low power electronics and design*, pages 154–159, New York, NY, USA, 2002. ACM Press.

[365] Daler N. Rakhmatov and Sarma B. K. Vrudhula. An analytical high-level battery model for use in energy management of portable electronic systems. In *ICCAD '01: Proceedings of the 2001 IEEE/ACM international conference on Computer-aided design*, pages 488–493, Piscataway, NJ, USA, 2001. IEEE Press.

[366] Bhaskaran Raman and Kameswari Chebrolu. Censor networks: a critique of "sensor networks" from a systems perspective. *SIGCOMM Comput. Commun. Rev.*, 38(3):75–78, 2008.

[367] Nithya Ramanathan, Kevin Chang, Rahul Kapur, Lewis Girod, Eddie Kohler, and Deborah Estrin. Sympathy for the sensor network debugger. In *Proceedings of the 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys '05)*, pages 255–267, 2005.

[368] R. Rao and S. Vrudhula. Battery optimization vs energy optimization: which to choose and when? In *ICCAD '05: Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*, pages 439–445, Washington, DC, USA, 2005. IEEE Computer Society.

[369] V. Rao, N. Navet, G. Singhal, A. Kumar, and G.S. Visweswaran. Battery aware dynamic scheduling for periodic task graphs. *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, pages 8 pp.–, April 2006.

[370] Theodore S. Rappaport. *Wireless Communications – Principles and Practice*. Prentice Hall, Upper Saddle River, NJ, USA, 2002.

[371] Srivaths Ravi, Anand Raghunathan, and Srimat Chakradhar. Tamper resistance mechanisms for secure, embedded systems. In *VLSID '04: Proceedings of the 17th International Conference on VLSI Design*, page 605, Washington, DC, USA, 2004. IEEE Computer Society.

[372] Srivaths Ravi, Anand Raghunathan, Paul Kocher, and Sunil Hattangady. Security in embedded systems: Design challenges. *Trans. on Embedded Computing Sys.*, 3(3):461–491, 2004.

[373] Niels Reijers and Koen Langendoen. Efficient code distribution in wireless sensor networks. In *Proc. of the 2nd Int. Conf. on Wireless Sensor Networks and applications (WSNA 2005)*, 2003.

[374] Charles Reis, Ratul Mahajan, Maya Rodrig, David Wetherall, and John Zahorjan. Measurement-Based Models of Delivery and Interference in Static Wireless Networks. In *Proc. ACM SIGCOMM*, Pisa, Italy, September 2006.

[375] Olof Rensfelt, Frederik Hermans, Christofer Ferm, Per Gunningberg, and Lars-Åke Larzon. An interactive testbed for heterogeneous wireless sensor networks. In *Demo Proceedings of the 4th International Conference on Distributed Computing in Sensor Systems (DCOSS 2008)*, 2008.

[376] Matthias Ringwald and Kay Römer. Bitmac: A deterministic, collision-free, and robust mac protocol for sensor networks. In *Proceedings of the 2nd European Workshop on Wireless Sensor Networks (EWSN)*, pages 57–69, Istanbul, Turkey, January 2005.

[377] Matthias Ringwald, Kay Römer, and Andrea Vitaletti. Passive inspection of wireless sensor networks. In *Proceedings of the 3rd International Conference on Distributed Computing in Sensor Systems (DCOSS 2007)*, 2007.

[378] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[379] U. Rohm, M. M. Gaber, and Q. Tse. Enabling resource-awareness for in-network data processing in wireless sensor networks. In *Proceedings of the 19th Australasian Database Conference*, pages 95–102, January 2008.

[380] Manuel Roman and Roy H. Campbell. Gaia: enabling active spaces. In *Proceedings of the 9th workshop on ACM SIGOPS European workshop*, pages 229–234, New York, NY, USA, 2000. ACM.

[381] Manuel Román, Fabio Kon, and Roy H. Campbell. Reflective Middleware: From Your Desk to Your Hand. *IEEE Distributed Systems Online*, 2(5), 2001.

[382] Rodrigo Roman, Cristina Alcaraz, and Javier Lopez. A survey of cryptographic primitives and implementations for hardware-constrained sensor network nodes. *Mob. Netw. Appl.*, 12(4):231–244, 2007.

[383] K. Römer and M. Ringwald. Increasing the visibility of sensor networks with passive distributed assertions. In *REALWSN 2008*, 2008.

[384] Kay Römer. The lighthouse location system for smart dust. In *MobiSys '03: Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 15–30, New York, NY, USA, 2003. ACM Press.

[385] Kay Römer, Christian Frank, Pedro José Marrón, and Christian Becker. Generic role assignment for wireless sensor networks. In *EW11: Proceedings of the 11th workshop on ACM SIGOPS European workshop*, page 2, New York, NY, USA, 2004. ACM.

[386] Bor rong Chen, Geoffrey Peterson, Geoff Mainland, and Matt Welsh. Livenet: Using passive monitoring to reconstruct sensor network dynamics. In *Proceedings of the 4th IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS 2008)*, pages 79–98, Santorini Island, Greece, June 2008.

[387] Stanislav Rost and Hari Balakrishnan. Memento: A health monitoring system for wireless sensor networks. In *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON)*, pages 577–584, Reston, Virginia, USA, September 2006.

[388] A. Rowe, M. Berges, G. Bhatia, E. Goldman, R. Rajkumar, L. Soibelman, J. Garrett, and J. M. F. Moura. Sensor andrew: Large-scale campus-wide sensing and actuation. Technical report, Carnegie-Mellon University, Pittsburgh, PA, 2008. Available at project website: http://www.ices.cmu.edu/censcir/resources/SensorAndrew-Tech-Report.pdf.

[389] A. Rowe, R. Mangharam, and R. Rajkumar. RT-Link: A time-synchronized link protocol for energy- constrained multi-hop wireless networks. In *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON '06)*, pages 402–411, Reston, VA, USA, September 2006.

[390] L.B. Ruiz, J.M. Nogueira, and A.A.F. Loureiro. Manna: a management architecture for wireless sensor networks. *Communications Magazine, IEEE*, 41(2):116–125, Feb 2003.

[391] A.M. Sabatini, C. Martelloni, S. Scapellato, and F. Cavallo. Assessment of walking features from foot inertial sensing. *IEEE Transactions on Biomedical Engineering*, 53(3), 2005.

[392] N. Sadagopan, B. Krishnamachari, and A. Helmy. The ACQUIRE mechanism for efficient querying in sensor networks. *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pages 149–155, 2003.

[393] J. M. Sánchez-Matamoros, J. R. Martínez de Dios, and A. Ollero. Cooperative localization and tracking with a camera-based wsn. In *Accepted in IEEE Intl. Conf. on Mechatronics (ICM2009)*, Malaga, Spain, April 2009.

[394] S. Sanghani, T.X. Brown, S. Bhandare, and S. Doshi. Ewant: the emulated wireless ad hoc network testbed. *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, 3:1844–1849 vol.3, March 2003.

[395] Yogesh Sankarasubramaniam, Özgür B. Akan, and Ian F. Akyildiz. Esrt: event-to-sink reliable transport in wireless sensor networks. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 177–188, New York, NY, USA, 2003. ACM.

[396] A. Santanche, S. Nath, J. Liu, B. Priyantha, and F. Zhao. Senseweb: Browsing the physical world in real time. In *IPSN 2008*, 2008.

[397] Paolo Santi. Topology control in wireless ad hoc and sensor networks. *ACM Comput. Surv.*, 37(2):164–194, 2005.

[398] S. Santini, B. Ostermaier, and A. Vitaletti. First experiences using Wireless Sensor Networks for noise pollution minotoring. In *REALWSN 2008*, 2008.

[399] Kimaya Sanzgiri, Bridget Dahill, Brian Neil Levine, Clay Shields, and Elizabeth M. Belding-Royer. A secure routing protocol for ad hoc networks. In *ICNP '02: Proceedings of the 10th IEEE International Conference on Network Protocols*, pages 78–89, Washington, DC, USA, 2002. IEEE Computer Society.

[400] Sap white paper: Toward a european strategy for the future internet. rq 26175 (08/09). http://www.europeansoftware.org/documents/SAP_WP_FutureInternet.pdf, December 2008.

[401] A. Sathyanarayana, S. Nageswaren, H. Ghasemzadeh, R. Jafari, and J.H.L.Hansen. Body sensor networks for driver distraction identification. In *ICVES*, 2008.

[402] Olga Saukh, Robert Sauter, Matthias Gauger, Pedro José Marrón, and Kurt Rothermel. On boundary recognition without location information in wireless sensor networks. In *IPSN '08: Proceedings of the 7th international conference on Information processing in sensor networks*, pages 207–218, Washington, DC, USA, 2008. IEEE Computer Society.

[403] Chris Savarese, Jan M. Rabaey, and Koen Langendoen. Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks. In *ATEC '02: Proceedings of the General Track of the annual conference on USENIX Annual Technical Conference*, pages 317–327, Berkeley, CA, USA, 2002. USENIX Association.

[404] Stefano Savazzi and Umberto Spagnolini. Energy Aware Power Allocation Strategies for Multihop-Cooperative Transmission Schemes. *IEEE Journal on Selected Areas in Communications*, 25(2):318–327, February 2007.

[405] D. Savio and S. Karnouskos. Web-service enabledwireless sensors in soa environments. In *Proc. IEEE International Conference on Emerging Technologies and Factory Automation ETFA 2008*, pages 952–958, 15–18 Sept. 2008.

[406] A. Savvides, H. Park, and M.B. Srivastava. The n-Hop Multilateration Primitive for Node Localization Problems. *Mobile Networks and Applications*, 8(4):443–451, 2003.

[407] Andreas Savvides, Chih-Chieh Han, and Mani B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179, New York, NY, USA, 2001. ACM Press.

[408] Anna Scaglione, Dennis L. Goeckel, and J. Nicholas Laneman. Cooperative communications in mobile ad hoc networks. *IEEE Signal Processing Magazine*, 23(5):18–29, September 2006.

[409] Guntram Scheible, Dacfey Dzung, Jan Endresen, and Jan-Erik Frey. Unplugged but connected – design and implementation of a truly wireless real-time sensor/actuator interface. *IEEE Industrial Electronics Magazine*, 1(2):25–34, 2007.

[410] Peter Schramm, Edwin Naroska, Peter Resch, Jörg Platte, Holger Linde, Guido Stromberg, and Thomas Sturm. A service gateway for networked sensor systems. *IEEE Pervasive Computing*, 3(1):66–74, 2004.

[411] Curt Schurgers, Olivier Aberthorne, and Mani B. Srivastava. Modulation scaling for energy aware communication systems. In *Proc. International Symposium on Low*

Power Electronics and Design (ISLPED'01), pages 96–99, Huntington Beach, CA, August 2001.

[412] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari. Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks. *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 108–121, 2004.

[413] SensorBase website. http://sensorbase.org. [Online; accessed: 14.05.2009].

[414] SensorScope website. http://sensorscope.epfl.ch. [Online; accessed: 14.05.2009].

[415] Sergio D. Servetto and Guillermo Barrenechea. Constrained random walks on random graphs: routing algorithms for large scale wireless sensor networks. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 12–21, New York, NY, USA, 2002. ACM.

[416] Arvind Seshadri, Mark Luk, Adrian Perrig, Leendert van Doorn, and Pradeep Khosla. Scuba: Secure code update by attestation in sensor networks. In *WiSe '06: Proceedings of the 5th ACM workshop on Wireless security*, pages 85–94, New York, NY, USA, 2006. ACM.

[417] ISO/IEC JTC1 Study Group on Sensor Networks Report, March 2009.

[418] S. H. Shah, K. Chen, and K. Nahrstedt. Available bandwidth estimation in ieee 802.11-based wireless networks. In *Proceedings of the ISMA/CAIDA Workshop on Bandwidth Estimation*, December 2003.

[419] S.H. Shah, K. Chen, and K. Nahrstedt. Dynamic bandwidth management in single-hop ad hoc wireless networks. *ACM Mobile Networks and Applications*, 10(1):199–217, February 2005.

[420] A. Sharaf, Jonathan Beaver, Alexandros Labrinidis, and K. Chrysanthis. Balancing energy efficiency and quality of aggregate data in sensor networks. *The VLDB Journal*, 13(4):384–403, 2004.

[421] Mohamed A. Sharaf, Jonathan Beaver, Alexandros Labrinidis, and Panos K. Chrysanthis. Tina: a scheme for temporal coherency-aware in-network aggregation. In *MobiDe '03: Proceedings of the 3rd ACM international workshop on Data engineering for wireless and mobile access*, pages 69–76, New York, NY, USA, 2003. ACM.

[422] Chien-Chung Shen, Chavalit Srisathapornphat, and Chaiporn Jaikaeo. Sensor information networking architecture and applications. *IEEE Personal Communications*, 8(4):52–59, 2001.

[423] ZhengMing Shen and Johnson P. Thomas. Security and qos self-optimization in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 7(9):1138–1151, 2008.

[424] Sudhakar Shetty. System of systems design for worldwide commercial aircraft networks. In *ICAS 2008*, 2008.

[425] E. Shi and A. Perrig. Designing secure sensor networks. *Wireless Communications, IEEE*, 11(6):38–43, Dec. 2004.

[426] Victor Shnayder, Bor-rong Chen, Konrad Lorincz, Thaddeus R. F. Fulford Jones, and Matt Welsh. Sensor networks for medical care. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 314–314, New York, NY, USA, 2005. ACM.

[427] N. Shrivastava, R. Mudumbai U. Madhow, and S. Suri. Target tracking with binary proximity sensors: fundamental limits, minimal descriptions, and algorithms. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 251–264, New York, NY, USA, 2006. ACM.

[428] N. Shrivastava, S. Suri, and Tóth C. D. Detecting cuts in sensor networks. In *Proc. of International Conference on Information Processing in Sensor Networks (IPSN'05)*, pages 210–217. IEEE, 2005.

[429] L. Shu, C. Wu, Y. Zhang, J. Chen, L. Wang, and M. Hauswirth. Nettopo: Beyond simulator and visualizer for wireless sensor networks. In *Second International Conference on Future Generation Communication and Networking (FGCN 2008)*, Hainan, China, December 2008.

[430] M.L. Sichitiu and V. Ramadurai. Localization of Wireless Sensor Networks with a mobile beacon. In *Proc. of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2004.

[431] Jaspreet Singh, Upamanyu Madhow, Rajesh Kumar, Subhash Suri, and Richard Cagley. Tracking multiple targets using binary proximity sensors. In *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks*, pages 529–538, New York, NY, USA, 2007. ACM.

[432] Jeff Slipp, Changning Ma, Nagesh Polu, James Nicholson, Martin Murillo, and Sajid Hussain. Winter: Architecture and applications of a wireless industrial sensor network testbed for radio-harsh environments. *Communication Networks and Services Research, Annual Conference on*, 0:422–431, 2008.

[433] Sensor Network Platform Kit. www.btnode.ethz.ch/Projects/MICSSensorNetworkPlatformKit. [Online; accessed: 14.05.2009].

[434] L. Song and D. Hatzinakos. A cross-layer architecture of Wireless Sensor Networks for target tracking. *IEEE/ACM Transactions on Networking (TON)*, 15(1):145–158, 2007.

[435] A.-K.H. Souley and S. Cherkaoui. Advanced mobility models for ad hoc network simulations. In *Proceedings Systems Communications*, pages 50–55, 2005.

[436] Eduardo Souto, Germano Guimar aes, Glauco Vasconcelos, Mardoqueu Vieira, Nelson Rosa, Carlos Ferraz, and Judith Kelner. Mires: a publish/subscribe middleware for sensor networks. *Personal Ubiquitous Comput.*, 10(1):37–44, 2005.

[437] P. Spiess and S. Karnouskos. Maximizing the business value of networked embedded systems through process-level integration into enterprise software. In *Proc. 2nd International Conference on Pervasive Computing and Applications ICPCA 2007*, pages 536–541, 26–27 July 2007.

[438] Vivek Srivastava, James Neel, Allen B. MacKenzie, Rekha Menon, Luiz A. DaSilva, James E. Hicks, Jeffrey H. Reed, and Robert P. Gilles. Using game theory to analyze wireless ad hoc networks. *IEEE Communications Surveys and Tutorials*, 7(4), 2005. http://www.comsoc.org/pubs/surveys.

[439] Jessica Staddon, Dirk Balfanz, and Glenn Durfee. Efficient tracing of failed nodes in sensor networks. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 122–130, New York, NY, USA, 2002. ACM.

[440] Stage - 2D multiple-robot simulator. http://playerstage.sourceforge.net/index.php?src=stage. [Online; accessed: 14.05.2009].

[441] J. Stankovic, I. Lee, A. Mok, and R. Rajkumar. Opportunities and obligations for physical computing systems. *IEEE Computer*, 38(11):25–33, 2005.

[442] J. A. Stankovic, Tarek F. Abdelzaher, Chenyang Lu, Lui Sha, and J. C. Hou. Real-time communication and coordination in embedded sensor networks. *Proc. of the IEEE*, 91(7):1002–1022, 2003.

[443] T. Stathopoulos, J. Heidemann, and D. Estrin. A remote code update mechanism for wireless sensor networks. Technical Report CENS-TR-30, UCLA, Center for Embedded Networked Computing, Nov. 2003.

[444] Illya Stepanov, Jörg Hähner, Christian Becker, Jing Tian, and Kurt Rothermel. A Meta-Model and Framework for User Mobility in Mobile Networks. In *Proc. of the 11th Int. Conf. on Networking 2003 (ICON 2003)*, pages 231–238, 2003.

[445] Radu Stoleru, Tian He, John A. Stankovic, and David Luebke. A high-accuracy, low-cost localization system for wireless sensor networks. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 13–26, New York, NY, USA, 2005. ACM Press.

[446] Radu Stoleru, Pascal Vicaire, Tian He, and John A. Stankovic. Stardust: a flexible architecture for passive localization in wireless sensor networks. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 57–70, New York, NY, USA, 2006. ACM Press.

[447] Tony Sun, Nia-Chiang Liang, Ling-Jyh Chen, Ping-Chieh Chen, and Mario Gerla. Evaluating mobility support in zigbee networks. In *EUC*, pages 87–100, 2007.

[448] Swiss experiment - interdisciplinary environmental research. http://lsir-swissex.epfl.ch. [Online; accessed: 14.05.2009].

[449] David Tacconi, Oscar Mayora, Paul Lukowicz, Bert Arnrich, Cornelia Setz, Gerhard Troster, and Christian Haring. Activity and emotion recognition to support early diagnosis of psychiatric diseases. In *In Proceedings of 2nd Pervasive Health Conference. Tampere, Finland*, 2008.

[450] Andrew S. Tanenbaum and Maarten Van Steen. *Distributed Systems: Principles and Paradigms*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.

[451] Gilles Thonet and Marc Bruel. Zigbee - the journey toward mass market adoption. *ST Journal of Research Wireless Sensor Networks*, 4(1), May 2007.

[452] Seth Tisue and Uri Wilensky. Netlogo: A simple environment for modeling complexity. In *in International Conference on Complex Systems*, pages 16–21, 2004.

[453] Ben L. Titzer, Daniel K. Lee, and Jens Palsberg. Avrora: scalable sensor network simulation with precise timing. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 477– 482, Los Angeles, California, USA, April 2005.

[454] F. Tobagi and L. Kleinrock. Packet Switching in Radio Channels: Part II–The Hidden Terminal Problem in Carrier Se nse Multiple-Access and the Busy-Tone Solution. *IEEE Transactions on Communications*, 23(12):1417–1433, December 1975.

[455] F. Tobagi and L. Kleinrock. Packet Switching in Radio Channels: Part III–Polling and (Dynamic) Split-Channel Reservation Multiple Access. *IEEE Transactions on Communications*, 24(8):832–845, Aug 1976.

[456] Gilman Tolle and David Culler. Design of an application-cooperative management system for wireless sensor networks. In *Proceedings of the 2nd European Workshop on Wireless Sensor Networks (EWSN)*, pages 121–132, January 2005.

[457] David Touzet, Frédéric Weis, and Michel Banâtre. Sensing and filtering surrounding data: The persend approach. In *Proceedings of the Mobile and Ubiquitous Information Access Workshop*. Springer, Sep 2003.

[458] Technology Research for Independent Living - Falls Prevention. http://www.trilcentre.org/falls_prevention/falls_prevention.474.html. [Online; accessed: 14.05.2009].

[459] E. Trumpler, V. Krunic, and R. Han. NodeMD: Diagnosing node-level faults in remote wireless systems. In *MOBISYS '07*, San Juan, Puerto Rico, June 2007.

[460] Nicolas Tsiftes, Adam Dunkels, and Thiemo Voigt. Efficient sensor network reprogramming through compression of executable modules. In *Proceedings of the Fifth Annual IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks*, June 2008.

[461] Tunstall Fall Management System. http://www.tunstall.co.uk/assets/Literature/6_1_41falls_management_solutions_sheet.pdf. [Online; accessed: 14.05.2009].

[462] Tutornet Testbed, University of Southern California. http://enl.usc.edu/projects/tutornet. [Online; accessed: 14.05.2009].

[463] University of Bonn. *BonnMotion - a mobility scenario generation and analysis tool*, 2005. http://www.cs.uni-bonn.de/IV/bomonet/BonnMotion.htm.

[464] UrbanSensing. http://research.cens.ucla.edu/projects/2006/Systems/Urban_Sensing/. [Online; accessed 14.05.2009].

[465] U.S. Census Bureau International database. http://www.census.gov. [Online; accessed 14.05.2009].

[466] Nitin H. Vaidya, Jennifer Bernhard, V. V. Veeravalli, P. R. Kumar, and R. K. Iyer. Illinois wireless wind tunnel: a testbed for experimental evaluation of wireless networks. In *E-WIND '05: Proceedings of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis*, pages 64–69, New York, NY, USA, 2005. ACM.

[467] Stefan Valentin. ChSim – a wireless channel simulator for OMNeT++. TKN Simulation Workshop 2006, Technical University of Berlin, Germany, September 2006. Available at project website: http://wwwcs.upb.de/cs/chsim.

[468] A. Varga. Using the omnet++ discrete event simulation system in education. *IEEE Trans. on Education*, 42:372, 1983.

[469] András Varga. The omnet++ discrete event simulation system. In *Proceedings of the European Simulation Multiconference (ESM)*, pages 319–324, Prague, Czech Republic, June 2001.

[470] R. Vaughan. Massively multiple robot simulations in stage. *Swarm Intelligence*, 2(2):189–208, December 2008.

[471] Richard T. Vaughan, Brian P. Gerkey, and Andrew Howard. On Device Abstractions for Portable, Reusable Robot Code. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robot Systems (IROS 2003)*, pages 2121–2427, Las Vegas, USA, October 2003.

[472] Olivier Villon and Christine Lisetti. A user model of psycho-physiological measure of emotion. In *Proceedings of the 11th international conference on User Modeling (UM 2007)*, pages 319–323, 2007.

[473] N. Vlajic and D. Xia. Wireless sensor networks: to cluster or not to cluster? *World of Wireless, Mobile and Multimedia Networks, 2006. WoWMoM 2006. International Symposium on a*, pages 9 pp.–, June 2006.

[474] M. Wachs, J. I. Choi, J. W. Lee, K. Srinivasan, Z. Chen, M. Jain, and P. Levis. Visibility: a new metric for protocol design. In *SenSys 2007*, 2007.

[475] Carl A. Waldspurger and William E. Weihl. Lottery scheduling: Flexible proportional-share resource management. In *Operating Systems Design and Implementation (OSDI)*, Monterey, CA, USA, November 1994.

[476] Chieh-Yih Wan, Andrew T. Campbell, and Lakshman Krishnamurthy. Pump-slowly, fetch-quickly (psfq) : A reliable transport protocol for sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(4):862–872, April 2005.

[477] Chieh-Yih Wan, Shane B. Eisenman, and Andrew T. Campbell. Coda: congestion detection and avoidance in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 266–279, New York, NY, USA, 2003. ACM.

[478] Peng-Jun Wan, Khaled M. Alzoubi, and Ophir Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *Mob. Netw. Appl.*, 9(2):141–149, 2004.

[479] Hua Wang, Helmut Prendinger, and Takeo Igarashi. Communicating emotions in online chat using physiological sensors and animated text. In *Conference on Human Factors in Computing Systems (CHI '04), extended abstracts*, 2004.

[480] Yong Wang and B. Ramamurthy. Group rekeying schemes for secure group communication in wireless sensor networks. *Communications, 2007. ICC '07. IEEE International Conference on*, pages 3419–3424, June 2007.

[481] You-Chiun Wang, Chun-Chi Hu, and Yu-Chee Tseng. Efficient deployment algorithms for ensuring coverage and connectivity ofwireless sensor networks. In *WICON '05: Proceedings of the First International Conference on Wireless Internet*, pages 114–121, Washington, DC, USA, 2005. IEEE Computer Society.

[482] Yue Wang, Jie Gao, and Joseph S.B. Mitchell. Boundary recognition in sensor networks by topological methods. In *Proc. of the 12th Int. Conf. on Mobile Computing and Networking*, 2006.

[483] R.D. Ward and P.H. Marsden. Physiological responses to different web page designs. *Int. J. Hum.-Comput.*, 59 (1-2)(pp 199-212), 2003.

[484] Torben Weis, Helge Parzyjegla, Michael A. Jaeger, and Gero Mühl. Self-Organizing and Self-Stabilizing Role Assignment in Sensor/Actuator Networks. In *Proceddings of the 8th International Symposium on Distributed Objects and Applications (DOA 2006)*, pages 1807–1824. Springer, Oct 2006.

[485] Matt Welsh and Geoff Mainland. Programming sensor networks using abstract regions. In *Proc. of the 1st Symp. on Network Systems Design and Implementation*, pages 29–42, 2004.

[486] Ye Wen, Rich Wolski, and Chandra Krintz. Online prediction of battery lifetime for embedded and mobile devices. *Special Issue on Embedded Systems: Springer-Verlag Heidelberg Lecture Notes in Computer Science*, 2004.

[487] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal. Firefly-inspired sensor network synchronicity with realistic radio effects. In *Proceedings of 3rd international conference on Embedded networked sensor systems (SenSys'05)*, pages 142–153, San Diego, California, USA, November 2005.

[488] Geoffrey Werner-Allen, Pat Swieskowski, and Matt Welsh. Motelab: A Wireless Sensor Network testbed. In *Proceedings of the Fourth International Conference on Information Processing in Sensor Networks (IPSN'05), Special Track on Platform Tools and Design Methods for Network Embedded Sensors (SPOTS)*, 2005.

[489] Kamin Whitehouse, Chris Karlof, and David Culler. A practical evaluation of radio signal strength for ranging-based localization. *SIGMOBILE Mobile Computing and Communications Review*, 11(1):41–52, 2007.

[490] Kamin Whitehouse, Cory Sharp, Eric Brewer, and David Culler. Hood: a neighborhood abstraction for sensor networks. In *Proc. of the 2nd International Conference on Mobile Systems, Applications, and Services*, pages 99–110, 2004.

[491] Kamin Whitehouse, Feng Zhao, and Jie Liu. Semantic Streams: A Framework for Composable Semantic Interpretation of Sensor Data . In *Proceedings of the Third European Workshop on Wireless Sensor Networks (EWSN '06)*, pages 5–20. Springer, Feb 2006.

[492] Frank H. Wilhelm, Monique C. Pfaltz, and Paul Grossman. Continuous electronic data capture of physiology, behavior and experience in real life: towards ecological momentary assessment of emotion. *Interacting with Computers*, 18(2):171–186, March 2006.

[493] A. Willig. Recent and emerging topics in wireless industrial communications: A selection. *Industrial Informatics, IEEE Transactions on*, 4(2):102–124, May 2008.

[494] Alec Woo, Sam Madden, and Ramesh Govindan. Networking support for query processing in sensor networks. *Commun. ACM*, 47(6):47–52, 2004.

[495] Alec Woo, Terence Tong, and David Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 14–27, 2003.

[496] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, and J. Stankovic. Alarm-net: Wireless Sensor Networks for assisted-living and residential monitoring. Technical report, Department of Computer Science, University of Virginia, 2006.

[497] A. D. Wood and J. A. Stankovic. Denial of service in sensor networks. *IEEE Computer*, 2002.

[498] C.S. Wu and V. O. K. Li. Receiver-initiated busy-tone multiple access in packet radio networks. In *Proceedings of the ACM workshop on Frontiers in computer communications technology*, 1987.

[499] Weili Wu, Hongwei Du, Xiaohua Jia, Yingshu Li, and Scott C.-H. Huang. Minimum connected dominating sets and maximal independent sets in unit disk graphs. *Theor. Comput. Sci.*, 352(1):1–7, 2006.

[500] Y. Wu, P.A. Chou, Qian Zhang, K. Jain, Wenwu Zhu, and Sun-Yuan Kung. Network planning in wireless ad hoc networks: a cross-layer approach. *Selected Areas in Communications, IEEE Journal on*, 23(1):136–150, Jan. 2005.

[501] S. Xu and T. Saadawi. Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks? *IEEE Communications Magazine*, 39:130–137, 2001.

[502] Feng Xue and P. R. Kumar. The number of neighbors needed for connectivity of wireless networks. *Wirel. Netw.*, 10(2):169–181, 2004.

[503] Q. Xue and A. Ganz. Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 63(2):154–165, February 2003.

[504] A. Yang, R. Jafari, P. Kuryloski, S. Sastry S. Iyengar, and R. Bajcsy. Distributed segmentation and classification of human actions using a wearable sensor network. In *Workshop on Human Communicative Behavior Analysis, International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[505] Y. Yang, F. Huang, X. Ge, X. Zhang, X. Gu, M. Guizani, and H. Chen. Double sense multiple access for wireless ad-hoc networks. *Comput. Netw.*, 51(14):3978–3988, 2007.

[506] Y. Yang, J. Wang, and R. Kravets. Achievable bandwidth prediction in multihop wireless networks. Technical report, Department of Computer Science Technical Report, University of Illinois at Urbana-Champaign, December 2003.

[507] You-Dong Yao. An effective go-back-n arq scheme for variable-error-rate channels. *IEEE Transactions on Communications*, 43(1):20–23, January 1995.

[508] YARP Project website. http://eris.liralab.it/yarp/. [Online; accessed: 14.05.2009].

[509] Wei Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 3:1567–1576 vol.3, 2002.

[510] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12(3):493–506, 2004.

[511] Mohamed Younis, Moustafa Youssef, and Khaled Arisha. Energy-aware management for cluster-based sensor networks. *Comput. Netw.*, 43(5):649–668, 2003.

[512] Y. Yu, B. Krishnamachari, and V.K. Prasanna. Issues in designing middleware for wireless sensor networks. *IEEE Network*, 18(1):15–21, January/February 2004.

[513] Yang Yu, Loren J. Rittle, Vartika Bhandari, and Jason B. LeBrun. Supporting concurrent applications in wireless sensor networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, pages 139–152, 2006.

[514] Hung yu Wei, S. Ganguly, and R. Izmailov. Ad hoc relay network planning for improving cellular data coverage. *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, 2:769–773 Vol.2, Sept. 2004.

[515] J. Zambreno, D. Honbo, A. Choudhary, R. Simha, and B. Narahari. High-performance software protection using reconfigurable architectures. *Proceedings of the IEEE*, 94(2):419–431, Feb. 2006.

[516] Manel Guerrero Zapata. Secure ad hoc on-demand distance vector routing. *SIGMOBILE Mob. Comput. Commun. Rev.*, 6(3):106–107, 2002.

[517] D. Zeinalipour-Yazti, P. Andreou, P.K. Chrysanthis, and G. Samaras. Mint views: Materialized in-network top-k views in sensor networks. *Mobile Data Management, 2007 International Conference on*, pages 182–189, May 2007.

[518] H. Zhai and Y. Fang. Physical carrier sensing and spatial reuse in multirate and multihop wireless ad hoc networks. In *INFOCOM '06: Proc. of the 25th IEEE International Conference on Computer Communications*, pages 1–12, 2006.

[519] Honglei Zhang, Shuangqing Wei, Ganesh Ananthaswamy, and Dennis L. Goeckel. Adaptive Signaling Based on Statistical Characterizations of Outdated Feedback in Wireless Communications. *Proceedings of the IEEE*, 95(12):2337–2353, December 2007.

[520] Wensheng Zhang and Guohong Cao. Dctc: Dynamic convoy tree-based collaboration for target tracking in sensor networks. *IEEE Transactions on Wireless Communications*, September 2004.

[521] J. Zhao, R. Govindan, and D. Estrin. Computing aggregates for monitoring wireless sensor networks. *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pages 139–148, May 2003.

[522] Jerry Zhao and Ramesh Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys)*, pages 1–13, 2003.

[523] Qing Zhao and Ananthram Swami. A decision-theoretic framework for opportunistic spectrum access. *IEEE Wireless Communications*, 4(4):14–20, August 2007.

[524] Y.J. Zhao, R. Govindan, and D. Estrin. Residual energy scan for monitoring sensor networks. *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, 1:356–362 vol.1, Mar 2002.

[525] Yuhao Zheng and Nitin H. Vaidya. Repeatability of illinois wireless wind tunnel. Technical report, University of Illinois at Urbana-Champaign, 2008.

[526] Fan Ye Gary Zhong. Gradient broadcast: A robust data delivery protocol for large scale sensor networks. *ACM Wireless Networks (WINET)*, 11:285–298, 2005.

[527] Gang Zhou, Tian He, Sudha Krishnamurthy, and John A. Stankovic. Impact of radio irregularity on wireless sensor networks. In *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 125–138, New York, NY, USA, 2004. ACM.

[528] Gang Zhou, Tian He, Sudha Krishnamurthy, and John A. Stankovic. Models and solutions for radio irregularity in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 2(2):221–262, 2006.

[529] Z. Zhou and K.C. Yow. Geographic ad hoc routing security: Attacks and countermeasures. *Ad Hoc & Sensor Wireless Networks*, 1(3):235–253, march 2005.

[530] Sencun Zhu, Sanjeev Setia, and Sushil Jajodia. Leap+: Efficient security mechanisms for large-scale distributed sensor networks. *ACM Trans. Sen. Netw.*, 2(4):500–528, 2006.

[531] Zigbee alliance. http://www.zigbee.org. [Online; accessed: 14.05.2009].

[532] M. Zuniga, C. Avin, and B. Krishnamachari. Using heterogeneity to enhance random walk–based queries. *Journal of Signal Processing Systems, Springer*, 2008.

[533] Marco Zuniga and Bhaskar Krishnamachari. An analysis of unreliability and asymmetry in low-power wireless links. *ACM Trans. Sen. Netw.*, 3(2):7, 2007.

Index

R

S

T

U

V

W

Y

Z

European Commission

Research Roadmap on Cooperating Objects

Luxembourg: Office for Official Publications oft he European Communities

2009 - 310 pp. - 14.8 x 21 cm

ISBN 978-92-79-12046-6
DOI 10.2759/11566