EAI Endorsed Transactions
on Internet of Things
Research Article ICST.ORG

Managing Heterogeneous WSNs in Smart Cities: Challenges and Requirements

Trang Cao Minh[1,*], Boris Bellata[1], Simon Oechsner[1], Ruizhi Liao[1] and Miquel Oliver[1]

Universitat Pompeu Fabra, Barcelona

Abstract

The dramatic advances in wireless communications and electronics have enabled the development of Wireless Sensor Networks (WSNs). WSNs consist of many affordable and portable sensor nodes for collecting data from the environment. In this tutorial article, we address management requirements of WSNs through presenting some key management scenarios in the Smart Cities context, such as intelligent transportation systems, smart grids and smart buildings. The limited resources and heterogeneous characteristics of WSNs pose new challenges in network management, which include the presence of various faults, the difficulty in replacing and repairing a large number of sensor nodes, the existence of an uncertain topology, and the resource allocation. To cope with these challenges, we first discuss advantages and disadvantages of centralized and distributed management approaches and then discuss the benefit of a multilevel management schema. Next, we present in detail the specific features for a management system of WSN in Smart Cities context (WSN-iSC) such as lightweight, self-detection, self-configuration, sharing infrastructure, service monitoring, plug and play, context awareness and interoperability. Finally, we discuss several key enabling technologies for management systems, such as policy based and agent based approaches, as well as we introduce some middleware solutions.

This tutorial article aims to be a first reference for any reader interested in WSN-iSC management solutions. It provides an insightful and comprehensible introduction to the scenarios, requirements, open challenges, problems, key technologies and desired features that will shape future developments on this field, as well as it surveys the most relevant and recent works from the literature.

Received on 24 May 2016; accepted on 08 July 2016; published on 01 December 2016

Keywords: sensor networks, network management, resources allocation, self configuration, context aware

doi:10.4108/eai.1-12-2016.151710

1. Introduction

Cities are getting increasingly crowded. Many research groups, in both academia and industry, have recently put huge efforts to make cities smarter to ensure public safety, provide efficient transport, save energy, reduce expenses, and improve the quality of life. With advances in wireless communications and MEMS (Micro Electro Mechanical Systems), Smart Cities [1] [1] [2] [3] are becoming a reality. Three most commonly deployed Smart Cites' applications are shown in Figure 1 and can be summarized as follows:

- **Intelligent Transportation Systems (ITS).** Intelligent transportation systems are applications which apply advances in information and communication technologies with the goal to organize the traffic more efficiently, enhance safety and reduce CO_2 emissions in transport systems. They can be deployed in vehicles (e.g., car, train, ship, and air plane) and in infrastructures (e.g., roads, train stations, and gas stations).

- **Smart Grids (SG).** The growing world population has created a greater demand for energy while the limited amount of fossil fuels is diminishing.

*Corresponding author. Email: trangcm@gmail.com

[1] In [1], a city can be defined as 'smart' when investments in human and social capital and traditional (transport) and modern information and communication infrastructures fuel sustainable economic development and a high quality of life, with a wise management of natural resources, through participatory action and engagement.

Additionally, the power grids designed and deployed in the past are not able to cope with current and future needs. To resolve these problems, smarter electrical grids which use information and communication technologies to optimize the energy distribution, and to improve the efficiency and productivity of the energy usage are being developed. New smart grids can also help suppliers and consumers to monitor and control the energy usage and costs.

- **Smart Home, Smart Building - Home and Office automation Systems (HOS).** Home and office automation systems interconnect electric devices such as heaters, lights, air conditioners, TVs, computers, alarms, and cameras through a communication network, allowing them to be remotely controlled, monitored or accessed from any room in the building, as well as from any location in the world by Internet. They help people to optimize their living style, arrange the day-to-day schedule, secure a high living quality, and reduce the energy consumption bills.

Recently, there are numerous research projects aiming at the development of technologies for such cities, such as Open Cities [4] and Smart Santander [5]. In the Open Cities project [4], several European cities such as Amsterdam, Barcelona, Berlin, Helsinki, Paris are working on exploring Open and User Driven Innovation methodologies to the Public Sector in a scenario of Future Internet Services for Smart Cities [6]. The Smart Santander project [5] focuses on designing, deploying and validating an experimental research facility to support typical applications and services for a smart city. In most of Smart Cities projects, wireless sensor networks (WSNs) play an important role in building instrumented and interconnected urban environments.

WSNs are made up of small, low power, and low cost automated devices (i.e., sensor nodes), which have the capability of sensing, data processing, and wireless communication at an affordable cost. Given these capabilities, WSNs can be deployed in different environments [7]. WSNs have wide applications in a variety of areas from industry, military to medical, scientific. Examples of applications include habitat monitoring, structure monitoring, smart homes and offices, surveillance, intelligent transportation systems, and many others [8] [9] [10] [11] [12] [13]. Therefore, WSNs are one of the critical components of Smart Cities. For example, in HOS, wireless sensor nodes are used to monitor or detect temperature, light, gas leaks and fire. The output of these sensors can be used to adjust the operation of electric appliances at homes. In ITS, numerous sensors installed on a vehicle can detect obstacles, measure the speed of the leading vehicle, warn impending collisions to the driver and trigger the collision avoidance system when necessary. Infrastructure sensors including induction loops, video and image processing, and microwave radars can be installed on the road to monitor traffic conditions and detect traffic congestion. Several recent studies have examined the feasibility of using WSNs in ITS. Wang et al. in [11], designed and implemented EasiTia, an applicable and cost-effective system for acquiring pervasive traffic information based on WSNs. Recently, Bottero et al. [12] have installed and tested a WSN traffic monitoring system in the area of a logistic platform at the Turin's freight village in Italy. In SG, sensors can be embedded in metering devices, placed at both end-points and in the transport network, to monitor and control the energy usage and/or the waste in real time both locally and remotely. They help operators and consumers to manage their energy usage efficiently, reducing their energy bills and optimize delivery networks. In [14], Matteo et al. addressed the challenges related to the use of self powered sensor nodes as a smart water meters that provide information about the water usage or the water quality. In addition, the authors proved that the maintenance costs of smart water meters can be saved when harvested energy is used. Another example of using sensor networks in smart meters systems can be found in [15]. Dede et al. [15] evaluated the use of smart meters for monitoring the power quality of a distribution grid. Lastly, Valls et al. developed an on-demand data collection WSNs for gas and water smart meters [13] supporting range sectoring to reduce the effect of hidden nodes.

However, one of the biggest limitations of WSNs is usually the scarcity of resources. Sensor nodes are equipped with small batteries with low power capacity. Therefore, sensor nodes are prone to fail due to the battery depletion. These failures can seriously affect the efficiency and the accuracy of the services provided by WSNs. For example, if some sensor nodes on a road are broken, the information about the traffic state (e.g., number of cars or obstacles) on that road might be wrong. Then, more cars will use that road since they have received inaccurate information about traffic and potential traffic congestion. Therefore, it is essential to have a management system to monitor network operations and nodes' state with the aim to detect and repair faults automatically, such that the efficiency and the accuracy of services which WSNs provide are ensured.

In this article, we focus on the management challenges that WSNs have in a Smart Cities context. The rest of the article is organized as follows. First, Section 2 introduces some early works in management systems for WSNs. Then, we introduce some scenarios to show what are the requirements in management for WSNs in Smart Cities in Section 3. Section 4 presents

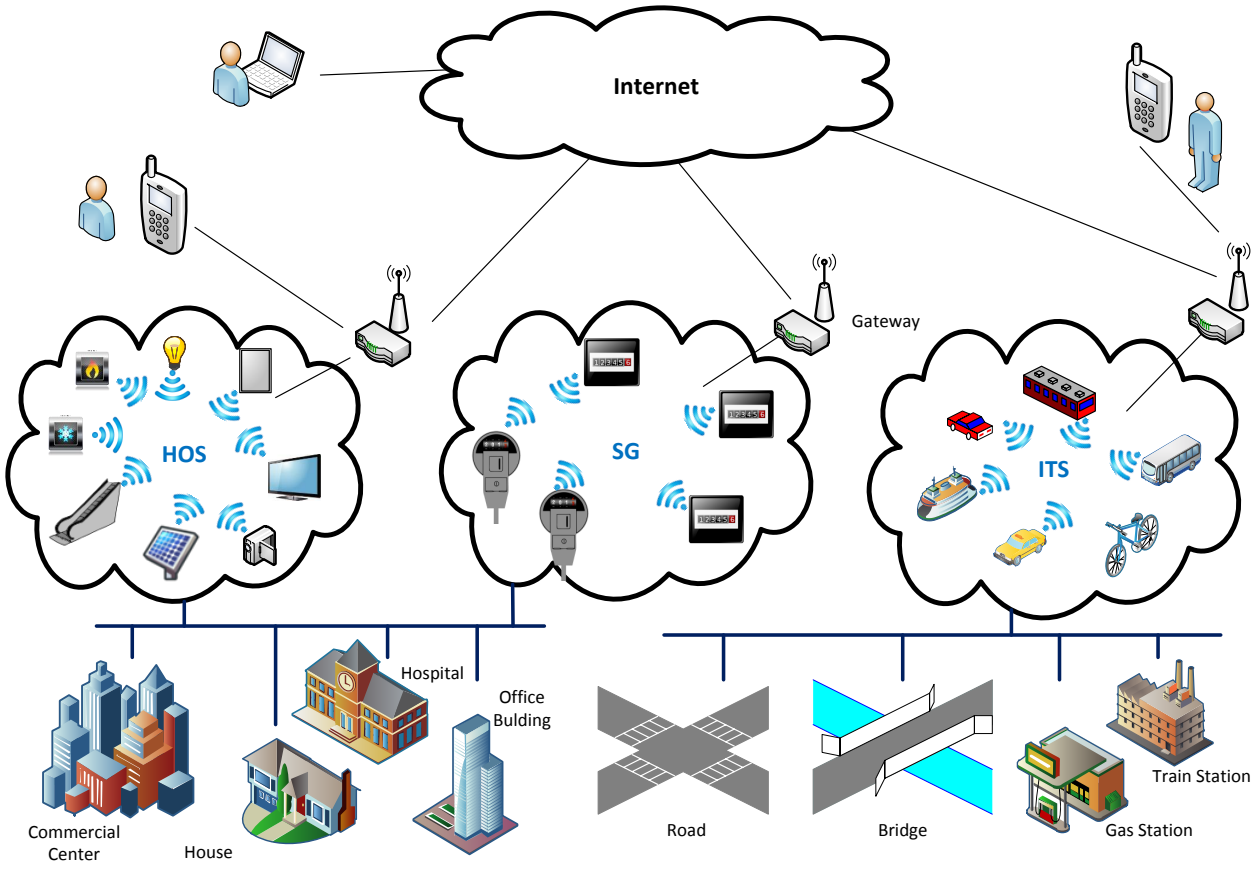

Figure 1. Smart City

the open challenges of WSN-iSC management systems. Section 5 outlines the most relevant functionalities that should be expected from a WSN-iSC management system. Section 6 presents the different architectural solutions that WSN-iSC management systems follows, as well as overviews the benefits and drawbacks of each one. Section 7 lists and discusses the desired features and properties that a WSN-iSC management system should have. We examine the key enabling technologies for network management in WSNs in Section 8. Finally, the conclusions are presented in Section 9.

2. Early Works

In traditional networks, Simple Network Management Protocol (SNMP) [16] is a standard protocol for managing networks. In SNMP, a manager station collects information from network agents in network elements. The structure of the management data in SNMP are described in Management Information Base (MIB). There are several reasons that make SNMP popular. First, it can manage a large range of devices. Second, it is a very flexible and extensible management protocol. Third, it is also proved to be good under poor network conditions. However, it requires transferring large amounts of management data between the manager and agents. This can potentially result in high energy and bandwidth consumption.

Ad Hoc Network Management Protocol ANMP [17] is an extended SNMP for wireless ad hoc networks, however, it can be used with certain types of WSNs. ANMP uses a hierarchical clustering mechanism for data collection to reduce the number of messages exchanged between the manager and agents (mobile nodes). In ANMP, the cluster head is responsible for collecting data from agents and forward them to the network manager. The nodes serving as cluster head change over time to adapt to node movements.

One of the earliest management systems for WSNs is MANNA [18]. It provides a general framework for policy-based management of WSNs. In MANNA, management services are executed by a set of functions. These functions are designed with a specific implementation for individual objectives in consideration of the unique features of WSNs. In order to provide the desired management services, MANNA defines policies that include conditions obtained from

WSN models that should be satisfied, for which specific functions are executed. The relationship among WSN models are defined in MIB which are updated frequently to adapt to network changes promptly. It is critical to determine the right time to query for management information and the right frequency for obtaining management information to ensure the accuracy of collected information while keeping low energy consumption.

In [19], Younis et al. divided the network into multiple clusters in which each cluster has a gateway node that organizes and manages network operations based on application requirements and the available energy in sensor nodes. However, their approach mainly focus on finding data relay route and arbitrating medium access.

Song et al. [20] presents another management system for WSNs based on Universal Plug and Play (UPnP) [21], the standard service discovery protocol for network management. It consists of three main components: control point, BOSS, and non-UPnP sensor nodes. The control point is a powerful device to support UPnP protocol. BOSS (Bridge Of the SensorS) is an UPnP agent, which is implemented in the base station and lies between the UPnP controllers and the non-UPnP sensor nodes to be managed. It contains the services of each sensor to provide them with a control point. It interprets and transfers messages between the sensor network and the control point.

Toller and Culer [22] proposed SNMS, a management system for WSNs which provides two mechanisms: query-based health data collection and persistent event logging. The query based health data collection mechanism allows users to collect the network data indicated in physical parameters to monitor the network health. The event logging mechanism allows nodes to store log events and send them to the users when they are requested. An improvement of SNMS based on Remote Procedure Call (RPC) mechanism is proposed in [23].

WinMS (Wireless Sensor Network Management System) [24] is an adaptive policy-based management system for WSNs. In WinMS, network states are monitored continuously to collect management data. When management parameters exceeds predefined thresholds, WinMS executes management tasks to reconfigure the network. In WinMS, individual sensor nodes can perform management functions locally based on the network state of their neighbors. The base station works as a central manager which stores and analyzes the global state of the network to detect interesting events and execute management maintenance.

3. WSNs Management Scenarios in Smart Cities

In this section, we present some scenarios in different Smart Cities applications to outline the requirements of WSN-iSC management systems.

3.1. Fault or Misbehavior

There are many faults or misbehaviors which can happen in WSNs in Smart Cities applications. In the following we will discuss two examples.

Scenario A: The water bill is wrong because consumption readings are not transmitted to the utility company due to problems at the meter, e.g., battery depletion or sensor broken, or errors on the data delivery path, e.g., network partition or network congestion.

Scenario B: The water bill is wrong because the utility company is receiving incorrect consumption readings caused by external attacks, misbehavior of the metering sensor, or errors in network protocols.

These two scenarios provide some different requirements for WSN-iSC management systems:

- First of all, a WSN-iSC management system must be able to determine what has caused the faults. This requires management tasks such as monitoring and fault tracking.
- To avoid unexpected effects when a fault occurs, a WSN-iSC management system needs to support fault predictability. In other words, it should be able to detect a fault before it occurs by analyzing and validating data including sensing data and network operation logs.
- Due to the existence of inevitable faults (e.g., in hardware, in software components, and in network links etc.), a WSN-iSC management system needs to detect these faults promptly and reconfigure the network operations to ensure the accuracy of the provided service.

3.2. Integration of new sensor nodes or new applications

During the network's lifetime, there might be the need for deploying new sensor nodes or new applications to replace the broken ones, to extend the network, to improve the network performance or to meet new users' requirements. The following are some examples of this situation:

Scenario A: A company wants to deploy a particular security application in its offices, which is located in a smart building. This application includes different types of sensors such as camera sensors, motion sensors and occupancy sensors to capture any unauthorized activity. Simultaneously, there might be other WSN

applications also deployed in the smart building such as the lighting system, the air conditioning system, and the alarm system. Taking advantage of existing resources in the building can reduce the deployment cost of the new user's security application. For example, it can utilize existing occupancy sensors of the lighting systems instead of deploying new ones.

Scenario B: In this scenario, sensor nodes powered by batteries are replaced by ones powered by solar energy in case sensor nodes are located in areas where sunlight is abundant such as green fields or roads. New energy harvesting sensor nodes can eliminate the inconvenience of replacing batteries, and also prolong WSN operational lifetime.

Scenario C: The deployment of a network may include several phases. In each phase, some new sensor nodes may be added to the network.

The management issues that arise in the above scenarios are as follows:

- Sensor nodes should be able to support multiple applications which can be owned by different users. A WSN-iSC management system needs to be able to allocate resources among applications, and also to ensure the privacy of each user.
- A WSN-iSC management system needs to have a power management mechanism to manage the harvested energy at the harvesting sensor nodes. This mechanism should be able to cooperate with the resource allocation function to align the workload with the energy availability at sensor nodes.
- The integration of new sensor nodes or new applications can require a code update process. Due to the large number of nodes in WSNs, manual updates are inefficient. Therefore, a WSN-iSC management system should have a remote configuration function.
- In order to ensure the compatibility between old sensor nodes and new ones, a WSN-iSC management system needs to update the network operations in which new sensor nodes can take part in, such as routing or the allocation of network resources for the running applications.

3.3. Quality of Service

Due to the variety of applications in WSNs, the required quality of service (QoS) varies greatly from application to application. For example, one of the QoS factors is the accuracy. In WSNs that provide information about the physical environment, the accuracy is measured by the discrepancy between the real world value and the provided results. However, in WSNs which are used to decide how to control actuators, the accuracy is measured by the discrepancy between the correct decision and the taken one. Moreover, different QoS factors such as delay and network lifetime may conflict by nature. Two scenarios are introduced to illustrate the conflict among QoS factors in a Smart Cities context.

Scenario A: In the fire detection system of a smart building, important events such as high temperature and the smoke occurrence need to be detected promptly. It requires a high data collecting rate which results in larger energy consumption, more network congestion and higher delays.

Scenario B: There are two WSN applications deployed on a road. The first application is used to detect the traffic congestion. The second one is to detect vehicles that cross a stop line while a red traffic light is on. There is a traffic congestion on the road. To keep live reports, camera sensors need to transmit information of the congestion (e.g., vehicle density, length of congestion, beginning and end of congestion) with a high rate to the sink, which affects the data traffic of the red light application. Information of some cars which violate traffic rules may be lost.

From the above scenarios, a WSN-iSC management system must consider the following requirements in order to ensure the required QoS:

- The WSN-iSC management system should define a QoS model for each application to identify the desired trade-off among QoS factors. It should also identify key QoS factors, if any, that influence the efficiency of the application. For example, the accuracy and the delay are more important for fire detection compared to other factors.
- The WSN-iSC management system should have a mechanism to monitor the QoS of running services to detect if the QoS of a service is met.
- When multiple applications are executed concurrently in a single WSN, the WSN-iSC management system should combine the QoS models of all applications, and generate a global QoS model to find the general trade-off in case the required QoS of all running applications can not be guaranteed.

3.4. Collaboration among WSNs

As mentioned above, there are multiple WSNs deployed in Smart Cities to support different applications. However, WSNs operate independently and belong to different authorities. Therefore, it would be efficient if different WSNs can cooperate to provide augmented services or to improve the network performance [25]. Some scenarios of the collaboration among WSNs are described as below:

Scenario A: A driver wants to find a parking place in a crowded area. The smart parking WSN and the traffic

monitoring WSN can collaborate to guide the driver to the most suitable empty parking place without trouble of traffic congestion.

Scenario B: Based on the collected information from the traffic monitoring WSN, the pollution monitoring WSN can adjust its data collecting rate correspondingly (e.g., the more traffic the higher collecting rate). This helps the pollution monitoring WSN to keep up-to-date information of pollution while optimizing energy consumption.

To support the collaboration among WSNs, there are new management requirements that a WSN-iSC management system needs to take into account.

- The WSN-iSC management system should be able to analyze and validate the collected data, or data requests received from external WSNs. Then, it should reallocate network resources to execute the received requests.

- The WSN-iSC management system needs to monitor and evaluate the effects of the collaboration with other WSNs on its own performance. It should be also able to use experience from previous similar collaboration requests in handling a new one.

4. Open Challenges

Based on the previous management scenarios, we identify the four main challenges that a WSN-iSC management system needs to solve. They are described in detail below:

- **Multiple types of failures.** In WSNs, faults happen more frequently than in other communication networks for many reasons. Firstly, sensor nodes have very limited resources. They are mainly equipped with a small power source (e.g., 2 AA batteries), which only allows them to be continuously active for few hours of continuous operation. In addition, batteries may be defective, hence, shortening node's lifetime. Therefore, sensor nodes are prone to fail due to the depletion of batteries. Secondly, WSNs can be deployed in heterogeneous environments such as houses, buildings, roads, and rivers. There are a lot of factors which can make sensor nodes or network links fail temporarily or permanently in those scenarios. For examples, nature disasters or traffic accidents can break connections or destroy sensor nodes in one area. Thirdly, WSNs can have a large number of sensor nodes in a small area. In other words, data traffic congestion may occur frequently if multiple nodes want to transmit packets simultaneously, which leads to packet losses. As has been pointed out, there are multiple different factors that can cause problems and failures in WSNs. Therefore, figuring out exactly and promptly their causes is extremely challenging for the WSN-iSC management system.

- **Difficult replacement and repair.** WSNs might be deployed in remote, unattended, or hostile environments, which makes difficult, expensive or sometimes impossible to replace or repair broken sensor nodes. In those conditions, potential failures should be identified or predicted before they occur. How sensor nodes are able to predict potential failures and find solutions to prevent them is still an open challenge.

- **Uncertain topology.** Depending on the application, the sensor network topology can be random or pre-determined. For example, in a smart house or a smart building, the location of the sensor nodes is specified. However, in forest fire detection systems, sensor nodes are deployed randomly. Moreover, after the deployment, there may exist a lot of factors that affect the WSN topology, including node faults, different wake up cycles or node movement. For example, node faults might result in broken links and the loss of network connectivity, or sensor nodes can wake up at different periods due to mis-configured or faulty network protocols. In some applications, the sink or sensor nodes are placed on movable objects such as a patient, a vehicle or an animal, resulting in a changing network topology. When the network topology is uncertain, keeping up-to-date network information is more costly since the WSN-iSC management system has to monitor frequently the network state. Moreover, management data could be also lost due to a change in the management data forwarding paths, which would result in the degradation of the efficiency of the WSN-iSC management system.

- **Resource allocation among heterogeneous sensor nodes.** Traditional WSNs are designed to support a single application that belongs to a single user. However, with the rapid development of MEMS technology, there are more differentiated types of sensor devices with different energy capacity and functionality. This results in the emergence of heterogeneous WSNs that consist of several different types of sensor nodes and, in addition, each sensor node may support multiple applications. For example, in a smart business building, the owner may deploy a WSN which supports multiple applications, including temperature and humidity monitoring, structure health monitoring and security alarms. Using a single

network to interconnect all nodes can reduce the deployment and maintenance costs since each node can run several applications. However, that situation also raises new challenges for the network management, such as how to allocate the network resources to different applications, how nodes collect and transmit measured data from different nodes and applications to the sink efficiently, and how to keep the energy consumption as low as possible. In addition, recent advances in technologies enable sensor nodes to collect and use energy from the environment [26], for example, light, differences in temperature, or linear motion instead of batteries. However, the availability of the harvested energy varies with time in a non deterministic manner. For example, the energy extracted from a solar panel depends on the maximum solar radiation and varies during a day. In addition, different nodes will have different harvesting opportunities. For example, sensor nodes placed at abundant sunlight areas can gather more energy than ones in shaded areas. Therefore, it is difficult to allocate tasks to the harvesting nodes since they do not have a stable energy source.

5. General Features

5.1. Definition of a WSN-iSC management system

Based on the management scenarios and challenges described previously, a next-generation WSN management system can be defined as: *A management system for WSNs in a Smart Cities context must be a **autonomic** system that keeps the network and the services that the network provides up and running smoothly with as **little human intervention** as possible, and consumes as **little resources and energy** as possible. It predicts potential problems, performs operations to avoid or locate them, and self-configures or suggests solutions to solve them. It also allows adjusting network operations and reprogramming nodes remotely. Finally, it supports allocating resources to the services offered by the network.*

The overall of network management for WSNs is to examine the way in that network resources are being used, and provide the necessary information for adjusting the operation of network so it optimizes the network usage and prolongs the network lifetime. According to the above definition, the detailed objectives of network management in WSNs in Smart Cities applications are:

- **Managing network resources and services**: monitor, control, update and report the status of sensor nodes and offered services.
- **Reliable services**: management systems for WSNs in Smart Cities applications should detect, diagnose, fix, predict and prevent faults and errors.
- **Limited human intervention**: WSN-iSC network management systems should enable sensor nodes to self manage as much as possible.
- **Prolong network lifetime**: allocate network resources. Network management should have the ability to choose a set of sensor nodes to offer a required service. It should also arrange and coordinate network resources to serve multiple services from different authorities.
- **Over-the-air update**: WSN-iSC network management systems should be able to reconfigure or reprogram network remotely.

These objectives are accomplished through basic management activities, each of that must be provided in an effective WSN-iSC management system:

- **Monitoring** is one of the most important management functions. It is responsible for collecting the information required by the management system to monitor the running status of the network, including the network topology, the remaining energy of the nodes in the network, and the QoS for the provided services, among others.
- **Resource Allocation** is necessary when multiple tasks, from different applications, running simultaneously in the same node and network. The resource allocation protocol is responsible for assigning network resources to different applications in order to ensure the QoS for the provided services while prolonging the network lifetime.
- **Fault Management** is responsible for the analysis of the causes and the search of solutions when a fault occurs.
- **Configuration** is used to reconfigure node's operation and update new code.

All these aspects are further described in next subsections.

5.2. Monitoring

Monitoring is one of the most important management functions. It is responsible for collecting the information required by the management system to monitor the running status of the network, including network topology, remaining energy of nodes in the network, quality of provided services, among others.

There are three basic monitoring mechanisms to be considered:

- **Periodic.** Sensor nodes should transmit information about its resources, e.g. the remaining battery level, operation logs of network protocols, or the number of hosting active applications to manager stations or the sink periodically. This would help to predict and diagnose possible problems and potential failures. Management information should be aggregated when transmitting to reduce management overhead. An example of the aggregation of management information is shown in Figure 2(a). In addition, transmission of management information has to face with a dilemma, as while the frequent transmission causes a lot of energy consumption, the infrequent one results in the late fault detection. Therefore, the period of time between two transmitted reports by each node must be optimized to provide the required information on time without overloading the network.

- **On demand.** A management system for WSNs in a Smart Cites context should allow to collect management data when it is needed. The Figure 2(b) illustrates a simple example of the on-demand monitoring mechanism. The sink detects a fault (e.g., the packet delivery rate suddenly drops below a threshold), hence it broadcasts management requests to the network to collect essential information to figure out what happened and why this issue has occurred.

- **Event based monitoring.** A sensor node should be able to send a notice to its responsible management node or the sink as soon as it detects a fault. It helps responsible management nodes and the sink to react promptly in case the detected fault is serious. As illustrated in Figure. 2(c), a sensor node detects that one of its neighbors may be broken because it has not received any information from this neighbor for a while. Once the sensor node notified its responsible management node of that unexpected event, it will investigate and evaluate the importance of that suspect node. If the suspect node has an important role in the communication paths or in some applications, the responsible management node will have to reconfigure the routes or reallocate resources to prevent a degradation in the network performance.

These monitoring mechanisms should work concurrently in the WSN-iSC management system to ensure there is no missing and unsolved problem. They should be able to cooperate in some management process. For example, the WSN-iSC management system detects a fault based on the collected information from the periodic monitoring mechanism. It can trigger the on demand monitoring mechanism to investigate what happened.

Some examples of monitoring approaches are proposed in [24, 27]. In [27], Liu et. al. proposed a two tier structure where nodes in the lower tier send status reports to nodes in the higher one. Each node at the higher tier makes local decisions based on the received data, and forwards its decisions towards the sink. Lee et al. in [24] presents a schedule-driven MAC protocol to collect and disseminate management data, to and from sensor nodes in a data gathering tree.

5.3. Resource Allocation

An efficient resource allocation schema is necessary when multiple tasks, from different applications, run simultaneously in the same node and network. It is responsible for assigning network resources to different applications in an optimal way in order to ensure the quality of provided services while prolonging the network lifetime. The first process of the resource allocation schema is validation, which includes the following mechanisms:

- **Access Verification.** This mechanism verifies that the users who request the task are authorized users.

- **Ability Validation.** This mechanism checks if the network, a sensor node or a group of nodes can satisfy the QoS required by the new task. It also checks if the new task affects the QoS of the existing tasks.

A simple illustration of the validation process is shown in Figure 3. When a node receives a new task, it verifies if if can execute that task based on its local resources and the task requirement. In case the node has not enough information to make decision, it sends message to other nodes to ask support. If the response from other nodes is positive, the node approve the task request. Otherwise, it denies. If the task is skipped, the node can send notifications about its decision or study how to process the similar tasks in future, then returns to the IDLE state.

In case the new task is approved, it becomes an active task. Before executing the new task, the WSN management system should be able to combine requirements from all active tasks. For example, consider that there is a task that collects temperature measurements if they are in the $[20, 30]°C$ range. Then, a new arriving task requests to collect temperature measurements if they are in the $[15, 25]°C$ range. In such case, the new task should report the temperature if it is in the $[15, 30]°C$ range. We refer to this new task as an aggregated task, as it aggregates in a single task the existing and the new one.

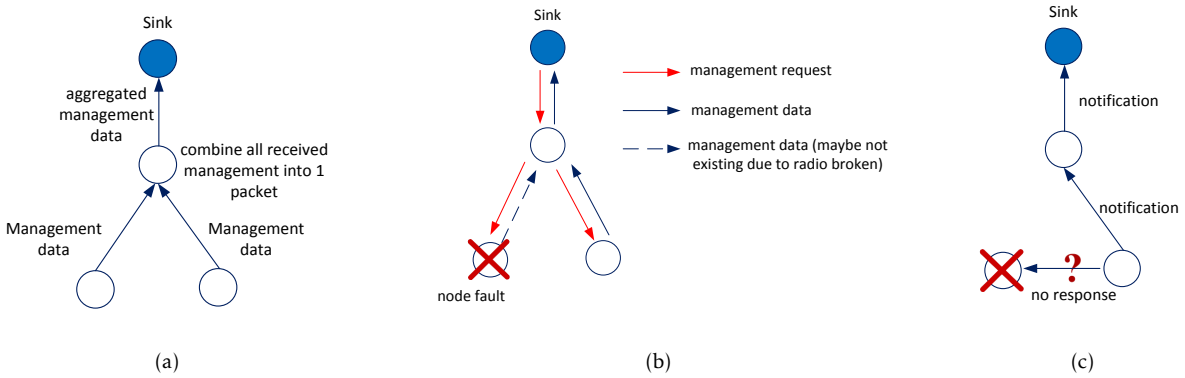

Figure 2. Monitoring mechanisms: (a) Periodic, (b) On demand, (c) Event based monitoring

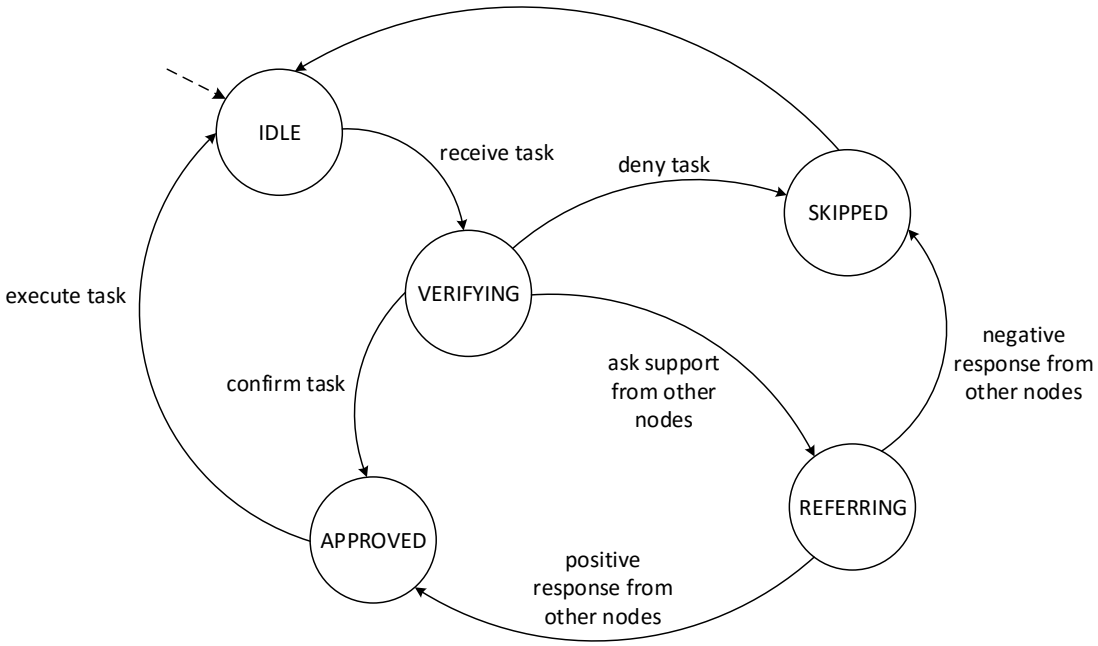

Figure 3. Validation process

Before a new task is implemented, it is necessary to have a scheduling mechanism to allocate active tasks in a way that sensor nodes are able to collect and disseminate its outputs intelligently to minimize the number of generated packets, and hence achieve bandwidth and energy efficiency. In [28], Minh et al. proposed a resource allocating mechanism where each node checks whether it can perform the requested operation based on its resources. Manager nodes coordinate tasks in groups of adjacent nodes based on the node's resources to avoid that multiple nodes execute the same task, hence, reducing the data duplication and the energy consumption.

Similarly, when a node is added or removed from the WSN, the WSN-iSC management system should be able to re-allocate network resources to ensure that they are used efficiently. In case the network performance is affected when applying new changes (e.g., the connectivity of network are lost when the removed nodes have important role in the routing path), the WSN-iSC management system should be able to suggest feasible solutions to resolve the problem.

5.4. Fault Management

The Fault Management system is responsible for the analysis of the causes and the search of solutions when

a fault occurs. When the Monitoring function detects a new fault, the WSN-iSC management system should perform the following steps:

- **Fault Evaluation.** This step carries out an initial assessment on the importance of the fault. It predicts the impact of the detected fault on the system operation and performance. If the fault is serious, it will continue investigating. Otherwise, it might record the fault to process it later, or simply ignore it. For example, if the sensing component of a sensor node is broken, this does not affect the data collection if there are other sensor nodes in the same area. However, these nodes might have the same fault in the future. Therefore, fault records can be used to identify the number of working nodes and send warnings when needed.

- **Fault Tracking.** This function is used to collect more information from the network to determine why the fault happened.

- **Fault Solution.** This function issues a solution to fix the fault. It should include a mechanism to predict the effect of the solution in the network operation and its performance.

One of the important tools for the fault management is the debugging tool. It is necessary to support both pre- and post-deployment debugging tools [29, 30]. The pre-deployment debugging tool allows for the prediction of possible failures, and also has the ability to handle them by simulating WSN operations. The post-deployment debugging tool is used to locale failures during the run time.

Since sensor nodes have very limited resources, it is costly to collect management data from all nodes in multi-hop networks. Therefore, there should be a simulation environment that reproduces the real network deployment (e.g. simulating network operations with the same number of nodes in a similar topology), which can be used to reproduce failures and evaluate the impact in a quick and cheap way.

5.5. Configuration

This function is used to reconfigure node's operation and update new code. There should be three levels of the node's configuration:

- Self-Configuring. Before any simple fault occurs, a sensor node should be able to self-configure its operation to avoid it. For example, harvesting nodes change their duty cycle according to the energy gathering rate to balance between the harvested energy and the consumption.

- Cooperative Configuring. Sensor nodes should be able to cooperate to configure their operation to avoid potential network failures. For example, in some cases, a node that has a low battery level can have an important role on a forwarding path. If that node is broken, it can cause a network partition. However, if it changes its duty cycle or if it decreases its transmission power, the lifetime of the node can be prolonged but the routing path can be affected negatively. Therefore, neighbor nodes can decide to change their routing tables to limit the amount of network traffic that goes through it. Additionally, they can cooperate to find the subset of nodes which are active in routing or in providing a specific service at a given time, while the others are inactive to reduce energy consumption [28, 31].

- Remote Configuring. The sink should be able to reconfigure the whole network when necessary. For example, it can change the duty cycle of all nodes, adjust the operating parameters of the network protocols to improve the network performance, or to deploy new code to update or change the network operation [32].

6. Management Architecture

One of most important aspects of the design a network management system is its architecture. Since the size of WSNs in Smart Cities applications can range from a small number to thousands of sensor nodes, a WSN-iSC management system should be scalable. It should work in both small and large networks. Additionally, it should support adding or removing nodes, protocols and applications easily and without affecting the on-going network operations and the perceived network performance.

6.1. Centralized approach

A centralized management server that processes the management data and take management decisions may be the best option for small networks. This central management server collects information from all sensor nodes and controls the entire WSN operation [29, 33]. Due to its abundant resources and the global knowledge of the WSN, it can perform complex management tasks and provide accurate management decisions. Complex management tasks are actions that require a high amount of resources and global knowledge of the network. For instance, controlling the network topology is a complex task. However, for large WSNs, it is difficult and costly to keep the management data from all the nodes in the network up to date. Firstly, sensor nodes cannot send management data frequently to the central server due to the high communication overheads

of the multi-hop forwarding. Secondly, transmitting management data frequently to the central server increases the traffic load of the nodes close to the sink, which can can cause network congestion and lead to high packet losses.

6.2. Distributed approach

Distributed management approaches are more suitable than centralized ones in large scale networks. Management decisions are taken by multiple manager stations [18, 34]. Each manager station controls part of the network (i.e., a group of nodes), and can cooperate with other manager stations if needed. However, the main disadvantage of the distributed approach is that manager stations do not have a global view of the network, as they only know the state of their respective subnetwork. Therefore, although their management decisions can be effective for their local subnetwork, they can affect negatively the operation of the overall network. For example, some nodes are turned off by a manager station in its subnetwork to optimize the resource usage. If those nodes happen to be the only connections to the rest of the network, the whole network will be severely affected. The cooperation among manager stations can improve somewhat this issue, but it may imply high overheads. For example, two manager stations can cooperate to decide which nodes are going to sleep without affecting the network connectivity. However, if the WSN has a lot of subnetworks, the number of management packets exchanged among manager stations will be high. Besides that, not all manager stations have rich power sources or strong processing capabilities, which means that the number of management functions or the complexity of management functions at those stations is limited.

6.3. Hybrid solution

Both centralized and distributed approaches have advantages and disadvantages. In order to cope with the heterogeneity of WSNs in Smart Cities applications, a hybrid management architecture could be designed to take advantages of both approaches for management of WSNs in Smart Cities. In a simple term, a hybrid management architecture consists of both centralized management server and manager stations to perform management functions based on the complexity and the cost required by these functions. For example, a recent work [28] proposed a multilevel management system for WSNs, in which every node, depending on its resources, participates in the management process at different levels. The approach allows some special nodes to manage a group of adjacent nodes. In small networks, these special nodes can be the sink nodes and the management approach becomes centralized. In case of large networks, these special nodes can be selected from nodes which have abundant resources. They can perform simple management tasks locally to reduce management traffic to the sink. Complex management tasks are performed by the sink, or an external server who has high processing capabilities.

A hybrid management architecture for management of WSNs in Smart Cities should have following features:

- **Reliable.** It can detect, handle, and isolate faults locally without affecting the functioning of rest of the network. In can also provide accurate management decisions due to the existence of the centralized management server.

- **Scalable.** It is easy to increase the size of network by adding new sensor nodes, without disturbing existing architecture.

- **Flexible.** According to the change of application requirements, the network topology, and the network resource, nodes can have different management roles. For example, when the remaining battery of a manager station is low, one of its neighbor can become a new manager station to ensure the functioning of management tasks in that area.

- **Effective.** Manager stations can be selected based on their network resource or their roles in network protocols (e.g. cluster head or parent node in routing tree). Therefore, the delay of handling management decisions and management overhead can be minimized.

Although the hybrid solution have many advantages, the design of a hybrid management architecture is complex. It requires to have an efficient choosing manager station algorithm. However, it is fortunately that there are a lot of clustering algorithms [15] which can be used to select manager stations, developed for WSNs. Another disadvantage of the hybrid solution is the management overhead. Exchanging management data can cause high traffic and energy consumption. Due to the limitation of resource in WSNs, a hybrid management architecture should be designed to ensure the trade-off between the management overhead and the efficiency of the management system. The number of exchanging management data should be minimized while it still ensures the accuracy of management decisions.

7. Desired Functionalities

In this section, we discuss some of the specific features of a WSN-iSC management system to cope with the challenges described above.

7.1. Lightweight

Since sensor nodes have limited resources, a WSN-iSC management system should be as lightweight as possible. The management functions and the management process should only occupy a small memory size. There should be a trade-off between the network traffic generated by a management process and the benefit derived from it.

7.2. Self-detection

There are a variety of faults in WSNs. Simple faults which are caused by hardware error or battery depletion should be detected locally by every sensor node. A couple of simple faults from different nodes can lead to a complex fault (e.g. network congestion or network partition). Complex faults can have a high probability to cause a degradation on the network performance. Therefore, sensor nodes should be able to collaborate to detect complex faults from simple faults.

7.3. Self-configuration

Every operations of sensor nodes should be optimized and able to adapt autonomously to the changes in resources and application requirements to prolong the network lifetime and prevent possible faults. For example, sensor nodes in the same sensing area can collect and transmit sensed data alternately. When a sensor node detects a fault, it should notify other nodes. Depending on the importance of the fault, sensor nodes should be able to adjust their operations to reduce negative effects caused by that fault. For example, if the sensing component of a sensor node is broken, it can have a more important role in the forwarding path since it does not need collect data from the environment. Therefore, other nodes can change their routing table to use that node as the forwarding node.

7.4. Sharing infrastructure

The deployment and maintenance of large WSNs with thousands of nodes require a high cost and huge effort. In case many WSNs are deployed in the same area, it would be efficient if they share their resources to support multiple applications from the different authorities. This is clearly seen in the two following examples:

- **Single application**. In a smart building, both the lighting system and the security system use occupancy sensors in rooms and corridors. In the lighting system, occupancy sensors are used to turn on/off the light depending on the presence of persons in the room. In the security system, they are used to start monitoring. Much of the same area is covered by both systems.
- **Single authority**. In a smart city, there are several organizations (e.g., police, highway agency, local city authorities) that need to deploy their own camera networks on the roads. However, these networks can cover the same areas and therefore, they may generate redundant information.

Therefore, it is beneficial to have a single infrastructure supporting multiple applications. The sharing infrastructure can include many different types of sensor nodes, in which some nodes support multiple applications. The management system of such an infrastructure should be able to allocate resources among applications to optimize the network performance. As the infrastructure can be shared by many authorities, it needs an access classification mechanism that assigns different privileges to different authorities to ensure the privacy.

7.5. Service Monitoring

The QoS of running services in WSNs should be monitored and evaluated periodically to detect whether it meets the predefined requirements. A WSN-iSC management system should also provide detailed information about the availability of the running services.

7.6. Context aware

As mentioned above, a WSN in a Smart Cities context should support multiple applications. Since each application has different requirements in terms of network resources, and both the application requirements and the network resources change over time, the network behavior should adapt to optimize its performance. During the network lifetime, there might be some situations that can be predicted before they happen. For example, there are more customers at commercial centers during weekends than weekdays. Then, to offer customers a comfortable shopping environment, the commercial centers' smart systems (e.g. lighting, air conditioning) may increase the operating power and the operating frequency autonomously every weekend. In such cases, a WSN-iSC management system should be able to use information of handled changes to process the current changes if they are similar. For example, it combines the total power consumption and the actual temperature in the floors of last weekend with current ones to adjust the air conditioners.

7.7. Plug and Play

Due to the heterogeneity of WSNs, management functions should be as independent as possible from hardware, network protocols and user applications. The same management function should be able to work

with different applications, different operating systems, different hardware and different network protocols. This would help to reduce the developing cost. In order to achieve this feature, management functions should be parameterizable and configurable. This allows to interface easily with other network protocols, hardware functions and different applications. In order to optimize the memory usage, management functions should be only added to a node when they are needed, and therefore, they should be able to be added or removed easily.

7.8. Interoperability

In order to enable the collaboration among different WSNs in Smart Cities applications, a WSN-iSC management system should support interoperability. It means the WSN-iSC management system is able to communicate and exchange data with ones of other WSNs. Data standards and public interfaces should be defined and unified among authorities of WSNs to facilitate the collaboration.

8. Recent Work & Key Technologies

We complete this tutorial describing a set of enabling technologies that are used for WSN network management such as policy based management approaches, agent based approached and middleware approaches.

8.1. Policy based management approaches

Policy-based management has emerged as a promising solution for the management of large-scale and heterogeneous networks. In policy based network management approaches, policies are defined as rules that govern the states and behaviors of the network system. Such policies are device independent and human friendly. Policies could be automatically updated to adapt to changes in the network state. Such automation is an essential requirement for large networks with frequent changes such as WSNs are. However, one disadvantage of policy based management approaches is its functional rigidity, that is, we can not add new management services to the system, unlike in the agent based management approaches.

Some early policy based management systems for WSNs are MANNA [18] and WinMS [24]. In MANNA and WinMS, policies specify management functions that should be executed if certain network conditions are met. Both of them use a central server to analyze the network state, and to execute corrective and preventive management actions according to predefined policies.

Other policy based management approaches for WSNs are described in [34], [35]. In [34], Cha et al. proposed an hierarchical framework in which the base station is responsible for interpreting high level management policies and distributing them to sensor nodes. These policies are then applied locally on each sensor node if its state matches the policy requirements. High level policies are defined in XML schema. They are distributed and interpreted to low level policy at sensor nodes. Le et al. [35] propose SRM, a hierarchical management architecture and policy-based network management paradigm for WSNs. There are three levels of policies in SRM: node level, cluster level, and base station level. At the node level, management policies consist of rules that require less resources and can be executed locally. The cluster level contains management polices that control the reliability of the cluster. The highest level, the base station level, include polices that control the entire network.

Zhu et al. [36] proposed Finger, an efficient policy based management system. Finger supports interpretation and enforcement of both obligation and authorization policies on all sensor nodes. Obligation policies are event-condition-action rules that perform an action in response to an event. Authorization policies define what resources or services a subject can access on a target sensor. Each sensor node has a Policy Decision Point (PDP) and a Policy Enforcement Point (PEP). The PDP is responsible for interpreting policies and making decisions. The PEP enforces the policy that is the result of PDP decision.

Policy based management systems can be also found in [37][38] [39] [40]. Matthys and Joosen [39] propose a policy driven middleware architecture to manage distributed sensor applications in a network infrastructure that consists of several sensor networks to offer services for different types of users. The proposed architecture supports two types of policies: i) functional policies, which are high level management goals, and ii) non functional policies, which are concrete goals that can be executed. Bourdenas and Sloman [40] describe the Starfish framework for specifying and dynamically managing policies in sensor nodes. It uses Finger2 which evolved from the original Finger system [36] to interpret and enforce policies.

Policies based management approaches are also used to manage some particular areas in WSNs. For example, policy based energy management systems are presented in [41] [42]. Bourdenas et al. [43] present a framework for autonomic task allocation in sensor networks based on Starfish [40]. Waterman et al. [44] have described the Peloton OS architecture that allows to distribute resource allocations to meet some desired policies. Misra and Jain [45] design a policy to activate the optimum number of sensor nodes such that the application fidelity is not affected based on the concepts of Markov Decision Processes (MDP).

8.2. Agent based management approaches

In agent based management approaches, a mobile agent is defined as a section of code that can distribute management tasks to be executed on nodes locally and returns the resulting data to the central manager [18]. The local processing can help to reduce the network bandwidth to the central manager server. However, some special nodes are required to act as agents and perform management tasks. These nodes should be placed intelligently to cover all the nodes in the network. In addition, the manager has to wait for the agent to visit the node in order to retrieve its status. This can cause delay. Some common examples of agent based management systems for WSNs are presented in [18] [46] [47] [48] [49].

Erdogan et al. [46] propose sectoral sweepers (SS) for managing a wireless sensor network. Each region of the network has a sectoral sweeper. The sectoral sweeper allows the central server to enable or disable tasks on nodes within a certain network region.

Agilla [47] is a mobile agent middleware designed to support self-adaptive applications in WSNs. It enables applications to be locally and autonomously self-adaptive by integrating the mobile agent and the tuple space programming models. Each sensor node maintains a tuple space that contains a set of predefined descriptors about that node. These tuple spaces can be accessed remotely. Each sensor node can be monitored by multiple agents. An agent can be cloned or moved across nodes. Agilla was designed for the TinyOS operating system [50].

There are several agent systems based on Java which are introduced in [48] [51] [49]. MASPOT [48] was a mobile agent system natively designed for the Sun SPOTs (Sun Small Programmable Object Technology) sensor devices [52]. Muldoon et al. [51] adopt Agent Factory Micro Edition (AFME), an intelligent agent framework for ubiquitous devices to sensor nodes. In [49] Haghighi and Cliff propose a novel middleware solution, which runs on Java (SE and ME) programming platforms for easy task distribution and data gathering integrated in a modulated architecture that supports the serving of multiple concurrent applications, dynamic reprogramming, good scalability, and multiple operational paradigms.

Rodriguez et al. [53] presents a new agent system that collects and analyzes data from WSNs deployed in the homes of elderly and dependent people to improve the health care and assistance services. In [53], the location of the people (i.e., patients and medical personnel) or assets (i.e., wheelchairs and lifters) are gathered from sensor devices. Then, the gathered information is analyzed by the PANGEA agent platform [54] and the Drool production rule system [55] to detect anomalous behaviors and determine what actions need to be executed.

Another mobile agent system is presented recently in [56] where sensor nodes delegate software agents (static or mobile) to collect and exchange data with their neighbor nodes. When the mobile agent migrates data from a big number of nodes, it needs the spatial data from a GIS (Geographic Information System) based host.

8.3. Middleware approaches

Middleware approaches add an additional logic layer in the firmware of motes in order to implement management services. These approaches provide a runtime environment that can support and coordinate multiple applications. They also provide mechanisms that optimize the system resources usage. In [57], Heinzelman et al. describe MILAN, a middleware that allows applications to specify their quality needs and adjusts the network operations to prolong network lifetime while still meeting these required quality needs.

TinyDB [58] provides users with a tool to query the network using SQL (Structured Query Language) languages. It collects and transmits sensing data from motes to the sink. Impala [59] is a middleware architecture that enables application modularity, adaptivity, and repair-ability in wireless sensor networks. It allows software updates to be received via the node's wireless transceiver and to be applied to the running system dynamically.

Mires [60] is a publish/subscribe middleware where the publish/subscribe service acts as a bridge between the local application and the communication components in a sensor node. Each node advertises the topics it can provide. The user application receives these topics and selects the desired topics to be monitored. After this, sensor nodes are able to publish the collected data of interest. Another public/subscribe middleware is presented in [61]. The middleware proposed in [61] provides application specific communication channels, and an approach to transform incoming sensor data to the desired data representation as well.

As mentioned previously, Agilla [47] is a mobile agent middleware that facilitates the user application deployment process. The RUNES middleware [62] is a component-based programming model where units of functionality and deployment are encapsulated in components. These components interact with each other through interfaces. RUNES supports dynamic reconfiguration that allows to upload and offload components and code dynamically.

Shah and Kumar [63] have proposed DReL, a middleware framework that provides mechanisms and data structures to allow support of applications with

different QoS requirements and optimization goal based on reinforcement learning and utility theory. In DRel, sensor nodes can decide whether to host an application task based on their capabilities and the utility of performing that task before. Ganz et al. [64] describe a middleware architecture that uses context information of sensors to supply a plug-and-play gateway and resource management framework for heterogeneous sensor networks.

There are several middleware systems that are proposed to support IoT applications recently. For example, SNPS [65] is an OSGi [66] based middleware that enable sensor nodes to be used and composed over the Internet in a simple and standardized way. Another example is MobIoT [67]. MobIoT is a service-oriented middleware that enables large-scale mobile participatory sensing. A new device can only register its service if it can provide new information that is not covered by the set of registered devices.

Mehrotra [68] presented SenSocial, a middleware that allows to link online social networks and sensor information in real time easily. The practicality of SenSocial was evaluated in two case studies. The first case study shows the social activity of the user on Facebook and links it to the physical context data acquired through mobile sensing in real-time. In the second application, SenSocial collects sensor data on the mobile of the user and sends it to the Web server. Then, the Web server generates a page corresponding to users momentary context extracted from the sensor data. In [69], Seeger et al. proposed MyHealthAssistant, a middleware for multiple medical applications on smart phone-like platform. MyHealthAssistant allows applications to define information types of theirs interest in a publish/subscribe manner. It uses a remote repository to transform the incoming sensor data into the desired data representation at run-time.

All the discussed approaches are summarized in the table 1.

9. Concluding Remarks

The heterogeneity of technologies and applications, as well as the specific requirements and limitations of WSNs, make necessary to deploy a smart management system to guarantee the correct operation and performance of the sensor networks. In this article, we have introduced a set of relevant management scenarios for WSNs, in the context of Smart Cities. Through those scenarios we have justified the requirements of a management system for WSNs in Smart Cities applications. We have then presented the objectives and challenges that a WSN-iSC management system needs to take into account. We have stated early works in the design of management systems for WSNs. Finally, we have discussed the appropriate management architecture , its basic functionalities and the overview of some key approaches proposed to resolve management problems in WSNs.

References

[1] A. Caragliu, C. Del Bo, and P. Nijkamp, "Smart Cities in Europe," *Journal of urban technology*, vol. 18, no. 2, pp. 65–82, 2011.

[2] K. Su, J. Li, and H. Fu, "Smart City and The Applications," in *Electronics, Communications and Control (ICECC), 2011 International Conference on*, pp. 1028–1031, Sept 2011.

[3] H. Chourabi, T. Nam, S. Walker, J. Gil-Garcia, S. Mellouli, K. Nahon, T. Pardo, and H. J. Scholl, "Understanding Smart Cities: An Integrative Framework," in *System Science (HICSS), 2012 45th Hawaii International Conference on*, pp. 2289–2297, Jan 2012.

[4] "Open Cities." http://opencities.net/.

[5] "Smart Santander." http://www.smartsantander.eu/.

[6] A. Domingo, B. Bellalta, M. Palacin, M. Oliver, and E. Almirall, "Public open sensor data: Revolutionizing smart cities," *Technology and Society Magazine, IEEE*, vol. 32, no. 4, pp. 50–56, 2013.

[7] C. Cano, B. Bellalta, A. Sfairopoulou, and M. Oliver, "Low energy operation in wsns: A survey of preamble sampling mac protocols," *Computer Networks*, vol. 55, no. 15, pp. 3351–3363, 2011.

[8] T. Naumowicz, R. Freeman, H. Kirk, B. Dean, M. Calsyn, A. Liers, A. Braendle, T. Guilford, and J. Schiller, "Wireless Sensor Network for habitat monitoring on Skomer Island," in *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, pp. 882–889, Oct 2010.

[9] M. Bocca, J. Toivola, L. Eriksson, H. J., and H. Koivo, "Structural Health Monitoring in Wireless Sensor Networks by the Embedded Goertzel Algorithm," in *Cyber-Physical Systems (ICCPS), 2011 IEEE/ACM International Conference on*, pp. 206–214, April 2011.

[10] D. Surie, O. Laguionie, and T. Pederson, "Wireless sensor networking of everyday objects in a smart home environment," in *Intelligent Sensors, Sensor Networks and Information Processing, 2008. ISSNIP 2008. International Conference on*, pp. 189–194, Dec 2008.

[11] R. Wang, L. Zhang, R. Sun, J. Gong, and L. Cui, "EasiTia: A Pervasive Traffic Information Acquisition System Based on Wireless Sensor Networks," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 2, pp. 615–621, 2011.

[12] M. Bottero, B. D. Chiara, and F. Deflorio, "Wireless Sensor Networks for Traffic Monitoring in A Logistic Centre," *Transportation Research Part C: Emerging Technologies*, vol. 26, no. 0, pp. 99 – 124, 2013.

[13] V. Valls, J. L. Sánchez, C. Cano, B. Bellalta, and M. Oliver, "Hierarchical range sectoring and bidirectional link quality estimation for on-demand collections in wsns," *Ad Hoc Networks*, vol. 11, no. 3, pp. 894–906, 2013.

[14] M. Mencarelli, M. Pizzichini, L. Gabrielli, and S. Squartini, "Self-powered sensor networks for water grids: challenges and preliminary evaluations," *Journal of Selected Areas in Telecommunications*, pp. 1–8, 2012.

Table 1. Summary

Type	Common characteristics	Application	Examples
Policy	Executes and behaves based on policies	Network Management	MANNA [18] WinMS [24] [34] SRM [35] Finger [36] [37] [38]
		Energy Management	[41] [42]
		Task Allocation	[39] [40], [43]
		Resource Allocation	[44]
		Active Node Selection	[45]
Agent	Uses section of code (mobile agent) to distribute tasks	Network Management	MANNA [18] [46]
		Deployment	Agilla [47]
		Java based system	[48] [51] [49]
		Data collection	[53] [56]
Middleware	Adds an additional logic layer to implement management services	QoS Support	MILAN [57] DRel [63] [64]
		Data collection	TinyDB [58]
		Deployment	Impala [59] Agilla [47] RUNES[62]
		Publish/Subscribe	[60] [61]
		IoT Applications	SNPS [65] MobIoT [67] SenSocial [68] MyHealthAssistant [69]

[15] A. A. Abbasi and M. Younis, "A Survey on Clustering Algorithms for Wireless Sensor Networks ," *Computer Communications*, vol. 30, no. 14â€"15, pp. 2826 – 2841, 2007.

[16] J. Case, "Management of High Speed Networks with The Simple Network Management protocol (SNMP)," in *Local Computer Networks, 1990. Proceedings., 15th Conference on*, pp. 195–199, Sep 1990.

[17] W. C. Nitin, W. Chen, N. Jain, and S. Singh, "ANMP: Ad hoc Network Network Management Protocol," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1506–1531, 1999.

[18] L. B. Ruiz, J. M. Nogueira, and A. A. F. Loureiro, "MANNA: A Management Architecture for Wireless Sensor Networks," *Communications Magazine, IEEE*, vol. 41, no. 2, pp. 116–125, 2003.

[19] M. Younis, M. Youssef, and K. Arisha, "Energy-aware Management for Cluster-based Sensor Networks," *Computer Networks*, pp. 649–668, 2003.

[20] H. Song, D. Kim, K. Lee, and J. Sung, "UPnP-Based Sensor Network Management Architecture," in *The Second International Conference on Mobile Computing and Ubiquitous Networking*, Apr. 2005.

[21] "Universal Plug and Play (UPnP)." http://tools.ietf.org/html/rfc6970.

[22] G. Tolle and D. Culler, "Design of an Application-Cooperative Management System for Wireless Sensor Networks," in *Wireless Sensor Networks, 2005. Proceedings of the Second European Workshop on*, pp. 121 – 132, jan.-2 feb. 2005.

[23] F. Yuan, W.-Z. Song, N. Peterson, Y. Peng, L. Wang, B. Shirazi, and R. LaHusen, "A Lightweight Sensor Network Management System Design," in *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*, pp. 288–293, 2008.

[24] W. L. Lee, A. Datta, and R. C. Oliver, "WinMS: Wireless Sensor Network-Management System, An Adaptive Policy-Based Management for Wireless Sensor Networks," tech. rep., School of Computer Science & Software Engineering, The University of Western Australia, 2006.

[25] S. Pal, S. Oechsner, B. Bellalta, and M. Oliver, "Performance optimization of multiple interconnected heterogeneous sensor networks via collaborative information sharing," *Journal of Ambient Intelligence and Smart Environments*, vol. 5, no. 4, pp. 403–413, 2013.

[26] S. Sudevalayam and P. Kulkarni, "Energy Harvesting Sensor Nodes: Survey and Implications," *Communications Surveys Tutorials, IEEE*, vol. 13, no. 3, pp. 443–461, 2011.

[27] C. Liu and G. Cao, "Distributed Monitoring and Aggregation in Wireless Sensor Networks," in *INFOCOM, 2010 Proceedings IEEE*, pp. 1–9, 2010.

[28] T. C. Minh, B. Bellalta, and M. Oliver, "DISON: A Self-organizing Network Management Framework for Wireless Sensor Networks," in *ADHOCNETS*, 2012.

[29] N. Ramanathan, K. Chang, R. Kapur, L. Girod, E. Kohler, and D. Estrin, "Sympathy for The Sensor Network Debugger," in *Proceedings of the 3rd international conference on Embedded networked sensor systems*, SenSys '05, (New York, NY, USA), pp. 255–267, ACM, 2005.

[30] Z. Chen and K. Shin, "Post-Deployment Performance Debugging in Wireless Sensor Networks," in *Real-Time Systems Symposium, 2009, RTSS 2009. 30th IEEE*, pp. 313–322, 2009.

[31] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-configuring Sensor Networks Topologies," *Mobile Computing, IEEE Transactions on*, vol. 3, no. 3, pp. 272–285, 2004.

[32] A. Hagedorn, D. Starobinski, and A. Trachtenberg, "Rateless Deluge: Over-the-Air Programming of Wireless Sensor Networks Using Random Linear Codes," in *Proceedings of the 7th international conference on Information processing in sensor networks*, IPSN '08, (Washington, DC, USA), pp. 457–466, IEEE Computer Society, 2008.

[33] W. Zhao, Y. Liang, Q. Yu, and Y. Sui, "H-WSNMS: A Web-Based Heterogeneous Wireless Sensor Networks Management System Architecture," in *Proceedings of the 2009 International Conference on Network-Based Information Systems*, NBIS '09, (Washington, DC, USA), pp. 155–162, IEEE Computer Society, 2009.

[34] S.-H. Cha, J.-E. Lee, M. Jo, H. Y. Youn, S. Kang, and K.-H. Cho, "Policy-Based Management for Self-Managing Wireless Sensor Networks," *IEICE Transactions*, pp. 3024–3033, 2007.

[35] T. Le, W. Hu, S. Jha, and P. Corke, "Design and Implementation of a Policy-based Management System for Data Reliability in Wireless Sensor Networks," in *Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on*, pp. 762 –769, oct. 2008.

[36] Y. Zhu, S. L. Keoh, M. Sloman, E. Lupu, Y. Zhang, N. Dulay, and N. Pryce, "Finger: An Efficient Policy System for Body Sensor Networks," in *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, pp. 428–433, Sept 2008.

[37] Y. Zhu, S. L. Keoh, M. Sloman, and E. Lupu, "A Lightweight Policy System for Body Sensor Networks," *Network and Service Management, IEEE Transactions on*, vol. 6, pp. 137 –148, september 2009.

[38] Z. Wenbo and X. Haifeng, "A Policy Based Wireless Sensor Network Management Architecture," in *Intelligent Networks and Intelligent Systems (ICINIS), 2010 3rd International Conference on*, pp. 552 –555, nov. 2010.

[39] N. Matthys and W. Joosen, "Towards Policy-based Management of Sensor Networks," in *Proceedings of the 3rd International Workshop on Middleware for Sensor Networks*, MidSens '08, (New York, NY, USA), pp. 13–18, ACM, 2008.

[40] T. Bourdenas and M. Sloman, "Starfish: Policy Driven Self-management in Wireless Sensor Networks," in *Proceedings of the 2010 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems*, SEAMS '10, (New York, NY, USA), pp. 75–83, ACM, 2010.

[41] X. Jiang, J. Taneja, J. Ortiz, A. Tavakoli, P. Dutta, J. Jeong, D. Culler, P. Levis, and S. Shenker, "An Architecture for Energy Management in Wireless Sensor Networks," *SIGBED Rev.*, vol. 4, pp. 31–36, July 2007.

[42] V. Sharma, U. Mukherji, V. Joseph, and S. Gupta, "Optimal Energy Management Policies for Energy Harvesting Sensor Nodes," *Wireless Communications, IEEE Transactions on*, vol. 9, pp. 1326–1336, April 2010.

[43] T. Bourdenas, K. Tei, S. Honiden, and M. Sloman, "Autonomic Role and Mission Allocation Framework for Wireless Sensor Networks," in *Self-Adaptive and Self-Organizing Systems (SASO), 2011 Fifth IEEE International Conference on*, pp. 61–70, Oct 2011.

[44] J. Waterman, G. W. Challen, and M. Welsh, "Peloton: Coordinated Resource Management for Sensor Networks," in *Proceedings of the 12th Conference on Hot Topics in Operating Systems*, HotOS'09, (Berkeley, CA, USA), pp. 9–9, USENIX Association, 2009.

[45] S. Misra and A. Jain, "Policy Controlled Self-configuration in Unattended Wireless Sensor Networks," *Journal of Network and Computer Applications*, vol. 34, no. 5, pp. 1530 – 1544, 2011. Dependable Multimedia Communications: Systems, Services, and Applications.

[46] A. Erdogan, E. Cayirci, and V. Coskun, "Sectoral Sweepers for Sensor Node Management and Location Estimation in Ad Hoc Sensor Networks," in *Proceedings of the 2003 IEEE Conference on Military Communications - Volume I*, MILCOM'03, (Washington, DC, USA), pp. 555–560, IEEE Computer Society, 2003.

[47] C.-L. Fok, G.-C. Roman, and C. Lu, "Agilla: A Mobile Agent Middleware for Self-adaptive Wireless Sensor Networks," *ACM Trans. Auton. Adapt. Syst.*, vol. 4, pp. 16:1–16:26, July 2009.

[48] R. Lopes, F. Assis, and C. Montez, "MASPOT: A Mobile Agent System for Sun SPOT," in *Autonomous Decentralized Systems (ISADS), 2011 10th International Symposium on*, pp. 25–31, March 2011.

[49] M. Haghighi and D. Cliff, "Multi-agent Support for Multiple Concurrent Applications and Dynamic Data-Gathering in Wireless Sensor Networks," in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on*, pp. 320–325, July 2013.

[50] P. Levis, "TinyOS 2.0 Overview." http://www.tinyos.net/dist-2.0.0/tinyos-2.0.0/doc/html/overview.html.

[51] C. Muldoon, G. O'Hare, M. O'Grady, and R. Tynan, "Agent Migration and Communication in WSNs," in *Parallel and Distributed Computing, Applications and Technologies, 2008. PDCAT 2008. Ninth International Conference on*, pp. 425–430, Dec 2008.

[52] E. Arseneau, R. Goldman, A. Poursohi, R. B. Smith, and J. Daniels, "Simplifying The Development of Sensor Applications," *Object-Oriented Programming Systems, Languages and Applications (OOPSLA 2006)*, 2006.

[53] S. Rodriguez, J. F. D. Paz, G. Villarrubia, C. Zato, J. Bajo, and J. M. Corchado, "Multi-agent information fusion system to manage data from a {WSN} in a residential home," *Information Fusion*, vol. 23, pp. 43 – 57, 2015.

[54] C. Zato, G. Villarrubia, A. Sanchez, I. Barri, E. Rubion, A. Fernandez, C. Rebate, J. Cabo, T. Alamos, J. Sanz, J. Seco, J. Bajo, and J. Corchado, "PANGEA - Platform for Automatic coNstruction of orGanizations of intElligent Agents," in *Distributed Computing and Artificial Intelligence*, vol. 151 of *Advances in Intelligent and Soft Computing*, pp. 229–239, Springer Berlin Heidelberg, 2012.

[55] "Drools. The Business Logic Integration Platform." http://www.drools.org/.

[56] N. Sahli, N. Jabeura, and M. Badra, "Agent-based framework for sensor-to-sensor personalization," *Journal of Computer and System Sciences*, vol. 81, no. 3, pp. 487 – 495, 2015. Special Issue on selected papers from the 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013).

[57] W. Heinzelman, A. Murphy, H. Carvalho, and M. Perillo, "Middleware to Support Sensor Network Applications," *Network, IEEE*, vol. 18, pp. 6–14, Jan 2004.

[58] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TinyDB: An Acquisitional Query Processing System for Sensor Networks," *ACM Trans. Database Syst.*, vol. 30, pp. 122–173, Mar. 2005.

[59] T. Liu and M. Martonosi, "Impala: A Middleware System for Managing Autonomic, Parallel Sensor Systems," in *Proceedings of the Ninth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP '03, (New York, NY, USA), pp. 107–118, ACM, 2003.

[60] E. Souto, G. Guimarães, G. Vasconcelos, M. Vieira, N. Rosa, C. Ferraz, and J. Kelner, "Mires: A Publish/Subscribe Middleware for Sensor Networks," *Personal Ubiquitous Comput.*, vol. 10, pp. 37–44, Dec. 2005.

[61] C. Seeger, K. Van Laerhoven, J. Sauer, and A. Buchmann, "A Publish/Subscribe Middleware for Body and Ambient Sensor Networks that Mediates between Sensors and Applications," in *Healthcare Informatics (ICHI), 2013 IEEE International Conference on*, pp. 199–208, Sept 2013.

[62] P. Costa, G. Coulson, R. Gold, M. Lad, C. Mascolo, L. Mottola, G. Picco, T. Sivaharan, N. Weerasinghe, and S. Zachariadis, "The RUNES Middleware for Networked Embedded Systems and its Application in a Disaster Management Scenario," in *Pervasive Computing and Communications, 2007. PerCom '07. Fifth Annual IEEE International Conference on*, pp. 69–78, March 2007.

[63] K. Shah and M. Kumar, "DReL: A Middleware for Wireless Sensor Networks Management using Reinforcement Learning Techniques," in *Proceedings of the 5th International Workshop on Middleware Tools, Services and Run-Time Support for Sensor Networks*, MidSens '10, (New York, NY, USA), pp. 1–7, ACM, 2010.

[64] F. Ganz, P. Barnaghi, F. Carrez, and K. Moessner, "Context-aware Management for Sensor Networks," in *Proceedings of the 5th International Conference on Communication System Software and Middleware*, COMSWARE '11, (New York, NY, USA), pp. 6:1–6:6, ACM, 2011.

[65] G. Di Modica, F. Pantano, and O. Tomarchio, "SNPS: An OSGi-Based Middleware for Wireless Sensor Networks," in *Advances in Service-Oriented and Cloud Computing* (C. Canal and M. Villari, eds.), vol. 393 of *Communications in Computer and Information Science*, pp. 1–12, Springer Berlin Heidelberg, 2013.

[66] O. Alliance, "Open Service Gateway Initiative, OSGi (2013)." http://www.osgi.org/.

[67] S. Hachem, A. Pathak, and V. Issarny, "Service-oriented Middleware for Large-scale Mobile Participatory Sensing," *Pervasive Mob. Comput.*, vol. 10, pp. 66–82, Feb. 2014.

[68] A. Mehrotra, V. Pejovic, and M. Musolesi, "Sensocial: a middleware for integrating online social networks and mobile sensing data streams," in *Proceedings of the 15th International Middleware Conference*, pp. 205–216, ACM, 2014.

[69] C. Seeger, K. Van Laerhoven, and A. Buchmann, "Myhealthassistant: An event-driven middleware for multiple medical applications on a smartphone-mediated body sensor network," *Biomedical and Health Informatics, IEEE Journal of*, vol. 19, pp. 752–760, March 2015.