INTERNATIONAL STANDARD

ISO/IEC 29182-4

First edition
2013-07-15

Information technology — Sensor networks: Sensor Network Reference Architecture (SNRA) —

Part 4: Entity models

Technologies de l'information — Réseaux de capteurs: Architecture de référence pour réseaux de capteurs —

Partie 4: Modèles des entités

Reference number
ISO/IEC 29182-4:2013(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 29182 consists of the following parts, under the general title *Information technology — Sensor networks: Sensor Network Reference Architecture (SNRA)*:

— *Part 1: General overview and requirements*

— *Part 2: Vocabulary and terminology*

— *Part 3: Reference architecture views*

— *Part 4: Entity models*

— *Part 5: Interface definitions*

— *Part 7: Interoperability guidelines*

The following part is under preparation:

— *Part 6: Applications*

Introduction

A wide range of applications has been proposed for sensor networks. In practice, however, sensor networks have been built and deployed for a relatively small number of applications. This is partly due to the lack of a business case for certain applications and partly due to technical challenges in building a non-trivial sensor network of reasonable complexity. The main reason for this impediment is the multi-disciplinary expertise – such as sensors, communications and networking, signal processing, electronics, computing, and cyber security – required to design a sensor network. Presently, the design process is so complex that one can leverage little from one sensor network design to another. It appears as if one has to start from almost scratch every time one wishes to design and deploy a sensor network. Yet, upon closer inspection, there are many commonalities in instantiations of sensor networks that realize various applications. These commonalities include similarities in the choice of network architecture and the entities/functional blocks that are used in the architecture.

The purpose of the ISO/IEC 29182 series is to

— provide guidance to facilitate the design and development of sensor networks,

— improve interoperability of sensor networks, and

— make sensor network components plug-and-play, so that it becomes fairly easy to add/remove sensor nodes to/from an existing sensor network.

The ISO/IEC 29182 series can be used by sensor network designers, software developers, system integrators, and service providers to meet customer requirements, including any applicable interoperability requirements.

The ISO/IEC 29182 series comprises seven parts. Brief descriptions of these parts are given next.

ISO/IEC 29182-1 provides a general overview and the requirements for the sensor network reference architecture.

ISO/IEC 29182-2 provides definitions for the terminology and vocabulary used in the reference architecture.

ISO/IEC 29182-3 presents the reference architecture from various viewpoints, such as business, operational, system, technical, functional, and logical views.

This part of ISO/IEC 29182 categorizes the entities comprising the reference architecture into two classes of physical and functional entities and presents models for the entities.

ISO/IEC 29182-5 provides detailed information on the interfaces among various entities in the reference architecture.

ISO/IEC 29182-6 provides detailed information on the development of International Standardized Profiles.

ISO/IEC 29182-7 provides design principles for the reference architecture that take the interoperability requirements into account.

There are no requirements for compliance in the ISO/IEC 29182 series. Users should ensure that the sensor nodes, and the related sensor network, are compliant with the application or deployment governing body.

Information technology — Sensor networks: Sensor Network Reference Architecture (SNRA) —

Part 4: Entity models

1 Scope

This part of ISO/IEC 29182 presents models for the entities that enable sensor network applications and services according to the Sensor Network Reference Architecture (SNRA).

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 29182-2, *Information technology — Sensor networks: Sensor Network Reference Architecture (SNRA) — Part 2: Vocabulary and terminology*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 29182-2 apply.

4 Abbreviated terms

3G	3rd Generation
4G	4th Generation
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
ICT	Information and Communication Technology
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IT	Information Technology
LBS	Location-Based Services
MAC	Medium Access Control
OSI	Open Systems Interconnection
PHY	Physical
PII	Personally Identifiable Information

QoS	Quality of Service
RF	Radio Frequency
RFID	Radio Frequency IDentification
SCM	Source Configuration Management
SDP	Service Discovery Protocol
SNRA	Sensor Network Reference Architecture
TEDS	Transducer Electronic Data Sheet

5 Overview

The purpose of this part of ISO/IEC 29182 is to provide basic information about and high-level models for various entities that comprise a sensor network. Entities can be roughly categorized into two classes, physical and functional. Physical entities are pieces of hardware and actual devices or components thereof that form the network, such as sensor nodes and gateways. For example, while a sensor node is a physical entity, so are any of the sensors in that node. A functional entity, on the other hand, represents a certain task that may be carried out on one or more types of physical entity. For example, data acquisition and collaborative information processing are both functional entities. While the former is carried out by the sensors, the latter is done "collaboratively" by sensor nodes, service providers, and users (or their machines, to be more precise). Routing and authentication are other examples of functional entities. More often than not, functional entities are pieces of code that run on physical entities.

Each entity model presented in this document is a description of the function/role of that entity. An attempt has been made to provide more detailed models for entities that are specific to sensor networks and typically not found in general-purpose communication networks. Examples of such physical entities include sensors and actuators. Similarly, more detailed models have been provided for functional entities such as data processing, self-localization, group management/clustering, collaborative information processing, and device management. A more detailed model may include an input-output relationship for what the entity does, some features of the entity that characterize its capabilities, and a taxonomy of various ways in which the entity may be implemented.

Figures 1 and 2 provide an overall view of the entities modelled in this document. Figure 1 is an amalgamation of Figure 3 in ISO/IEC 29182-1[1] and Figure 4 in ISO/IEC 29182-3[2]. It shows the physical entities that form a sensor network and how these entities are connected to each other. The blow-up part of the figure is borrowed from ISO/IEC 29182-3 and it shows the internal structure of a sensor node. It implies that actuator(s), although associated with sensor nodes, may not physically reside in sensor nodes. The rest of the figure comes from ISO/IEC 29182-1 and it depicts a more complex instantiation of a sensor network than the other cases presented in Figures 1 and 2 in ISO/IEC 29182-1. Figure 2 is the same as Figure 7 in ISO/IEC 29182-3. It has been reproduced in this document for ease of reference.

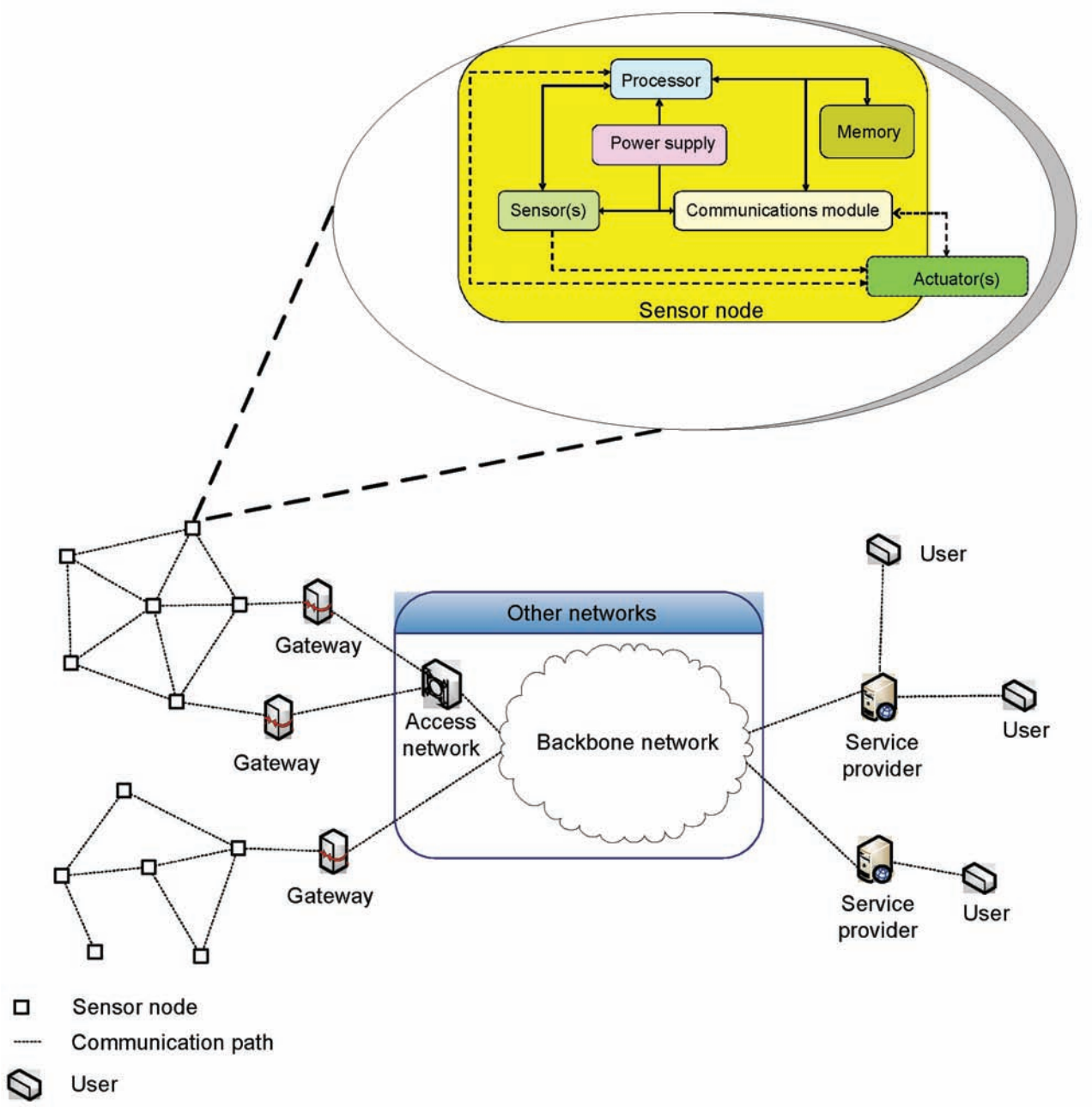

Figure 1 — Physical entities of a sensor network

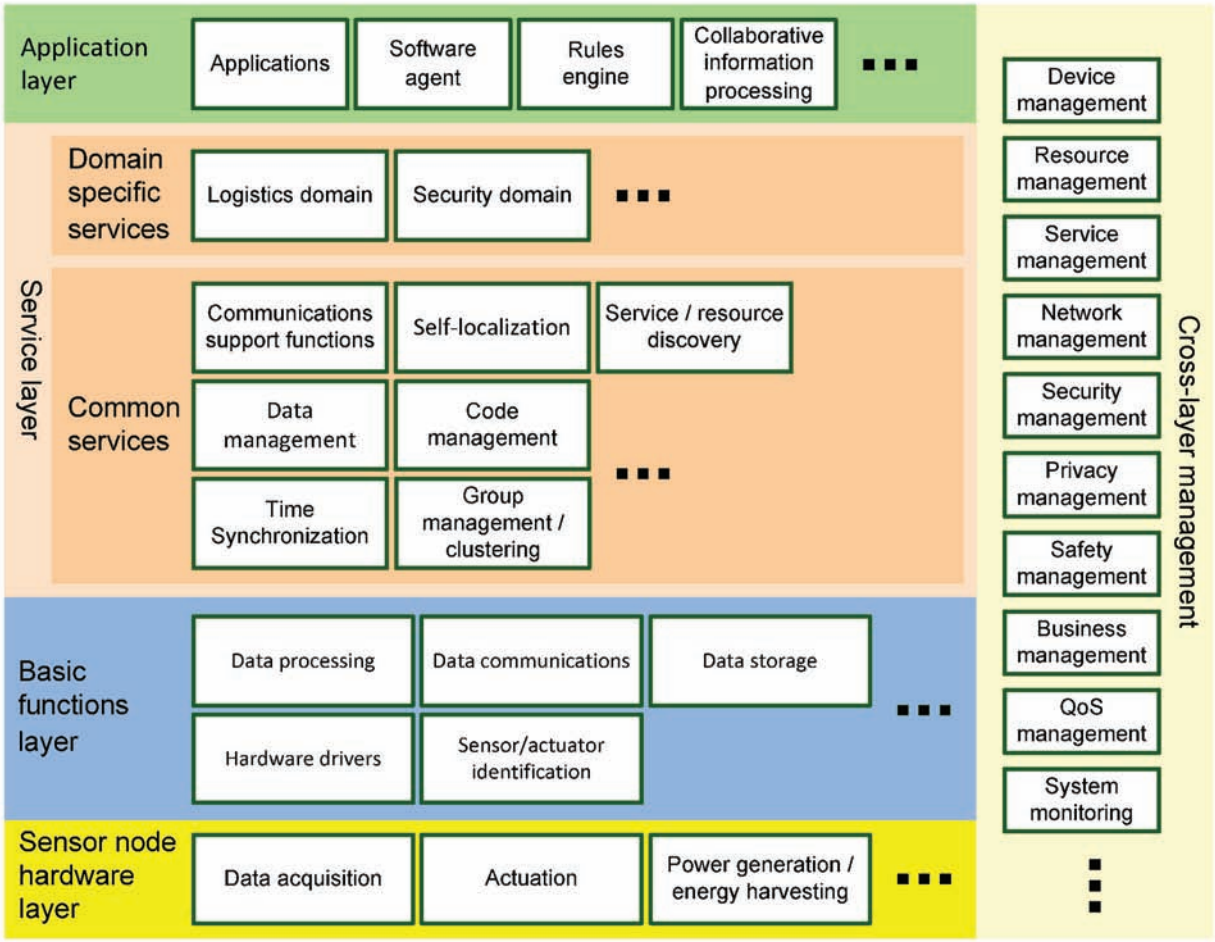

Figure 2 — Functional entities of a sensor network

The distinction between physical and functional entities in a sensor network and how they relate to each other can at times be confusing. Table 1 is an attempt to remedy this problem. It shows all the physical entities a functional entity could be associated with. The word "could" has been used here, because some physical entities may not be present in a given sensor network. For example, if there are no service providers in the overall architecture, which would be the case in a stand-alone sensor network, then one cannot say that there is an association between the collaborative information processing functional entity and service providers. In other words, the reader is urged to think of Table 1 as representative of certain possibilities, but not covering all possible ways of which entities might be present in the sensor network and how they might be configured.

Table 1 — Interrelationships between physical and functional entities in a sensor network

				Physical entities										
				Sensor nodes							Other networks			
				Sensors	Actuators	Communications module	Processor	Memory	Power supply	Gateways	Access networks	Backbone network	Service providers	Users
Functional entities	Sensor node hardware layer		Data acquisition	●										
			Actuation		●									
			Power generation / energy harvesting						●					
	Basic functions layer		Data processing	●			●	●						
			Data communications			●				●	●	●	●	●
			Data storage					●					●	●
			Hardware drivers	●	●		●							
			Sensor/actuator identification	●	●		●							
	Service layer	Common services	Communications support functions			●	●	●		●	●	●	●	●
			Self-localization	●		●	●	●						
			Service/resource discovery			●							●	●
			Data management				●	●					●	●
			Code management				●	●		●			●	●
			Time synchronization			●	●						●	●
			Group management / clustering			●	●	●					●	
			Domain specific services				●						●	
	Application layer		Applications				●						●	
			Software agent				●			●			●	●
			Rules engine				●						●	●
			Collaborative information processing	●			●	●					●	●
	Cross-layer management		Device management	●	●	●				●				
			Resource management	●	●	●	●	●					●	●
			Service management										●	●
			Network management			●	●	●					●	●
			Security management			●	●	●		●			●	●
			Privacy management											
			Safety management											
			Business management				●	●					●	●
			QoS management	●	●	●	●	●		●			●	●
			System monitoring	●	●	●	●	●	●					

6 Physical entities

6.1 Sensor nodes

6.1.1 Overview

As it was stated earlier and depicted in the upper part of Figure 1, a sensor node comprises several sub-entities whose models are presented next. Note that the actuator(s), if at all present, may not physically reside inside the sensor node.

6.1.2 Sensors

A sensor measures a physical attribute, such as temperature, humidity, or level of carbon monoxide in the air, and converts it into an electric voltage/current. This conversion may be direct or indirect. While in the former case the attribute is directly converted into an electric voltage/current, in the latter case the attribute is converted into a sequence of one or more intermediate attributes before finally getting converted into an electric voltage/current. For example, a thermometer may measure temperature and convert it into physical displacement of some object and then convert that displacement into an electric voltage/current. The sensor output voltage/current may be in analog or digital form. In the former case an analog to digital converter (ADC) is used to convert the analog electric voltage/current into a finite-length sequence of bits (binary digits) that constitutes a binary representation of the voltage/current.

Therefore, an appropriate model for a sensor with analog output is an input-output relationship that characterizes the conversion from the attribute being measured by the sensor into the output electric voltage/current. The relationship may occasionally be characterized through a mathematical formula or more frequently through a xy-plot. For example, Figure 3 shows the output voltage of a temperature sensor versus the input temperature. As the temperature increases, the output voltage decreases, which is an indication of a negative temperature coefficient.

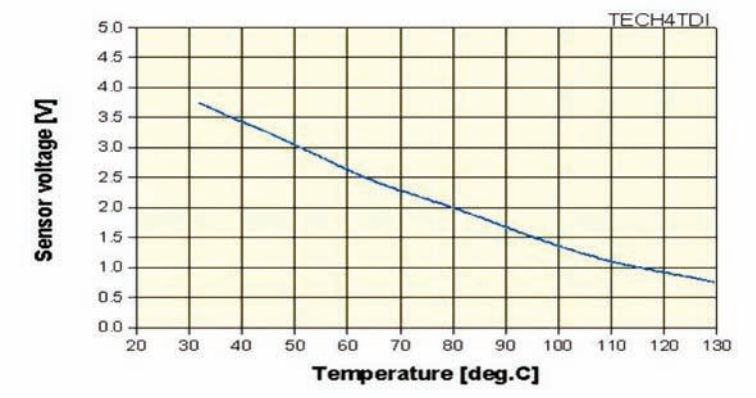

Figure 3 — Input-output relationship of a temperature sensor

On the other hand, an appropriate model for a sensor with digital output is a quantizer input-output plot or table. The former is a staircase xy-plot that represents the analog physical attribute on the horizontal axis and the analog value represented by the sensor binary output on the vertical axis. Figure 4 shows a quantizer input-output plot.

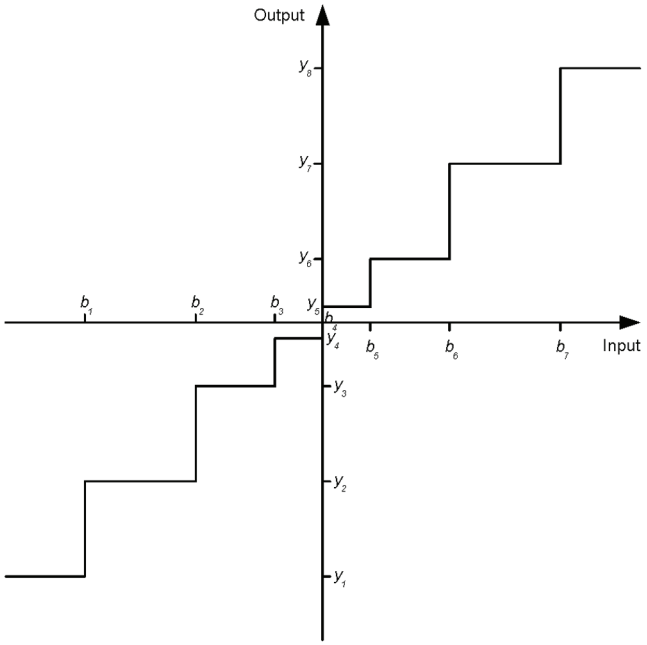

Figure 4 — Input-output relationship for an 8-level non-uniform quantizer

Alternatively, two tables may be used to characterize the same input-output relationship as well as to specify the binary codewords used by the quantizer. For example, any temperature between 18 and 18.1 °C may be represented by the binary codeword "10110101", and in turn 18.05 °C may be used as the representative value for that temperature range and the associated binary codeword. Tables 2 and 3 show, respectively, the operations of the encoder and decoder of the quantizer depicted in Figure 4.

Table 2 — The encoder for the quantizer shown in Figure 4

Quantizer Input Range	Output Binary Codeword
$(-\infty, b_1)$	000
$[b_1, b_2)$	001
$[b_2, b_3)$	010
$[b_3, b_4)$	011
$[b_4, b_5)$	100
$[b_5, b_6)$	101
$[b_6, b_7)$	110
$[b_7, \infty)$	111

Table 3 — The decoder for the quantizer shown in Figure 4

Input Binary Codeword	Quantizer Output Level
000	y_1
001	y_2
010	y_3
011	y_4
100	y_5
101	y_6
110	y_7
111	y_8

The models presented above are deterministic in nature and as such ignore presence of measurement noise in the sensor. A model that takes these aspects into account may be of the form

$$X = a\,S + N$$

where

S is either a deterministic or random variable representing the physical attribute being measured by the sensor,

a is a conversion/scaling factor,

N is a random variable representing measurement noise, and

X is the sensor output electric voltage/current.

When using a stochastic sensor model like the above, one needs to specify the conversion/scaling factor a and the probability distribution for additive noise N. When the physical attribute being measured is modelled by a random variable S, it is necessary to provide the joint probability distribution of S and the noise random variable N. Usually, but not always, S and N are assumed to be statistically independent. In case of a sensor with a digital output, the quantizer used by the sensor needs to be characterized per earlier discussion.

In case of all the models discussed above, it is also necessary to specify the range of input values – i.e. the range of values for the physical attribute – over which the sensor functions. Finally, in case of a sensor node with multiple sensors, a model needs to be provided for each sensor used in the node.

6.1.3 Actuators

Roughly speaking, an actuator functions in a manner that is the inverse of how a motion sensor functions. It takes an electric voltage/current, in analog or digital form, as input/command and causes some motion (translation, rotation), thereby moving or changing the orientation of the object being controlled by a certain amount. In many cases, the electric voltage/current is first converted to hydraulic pressure and it's the latter that causes motion. This parallels the discussion of direct or indirect conversion in the case of sensors.

It is straightforward to develop a deterministic model for an actuator following a line of thought similar to that presented in Subclause 6.1.2. An actuator with an analog input electric voltage/current is modelled by a xy-plot and a range of values for the input. The simplest way to model an actuator with digital input is through the use of a table that shows the motion values for all possible binary input strings. This would work for small-size tables. If the binary string is more than a few bits long, then a functional form would be the appropriate model. In such a case, the function operates on the decimal value represented by the binary input string to the actuator and converts that into motion.

Similarly, a stochastic model for an actuator would take the form

$$X = a\,s + N$$

where

- s is the deterministic input/command to the actuator,
- a is a conversion/scaling factor, and
- N is a random variable characterizing any randomness in actuator operation as the same input may not always result in exactly the same motion.

6.1.4 Communications module

A sensor node may have more than one mechanism for communicating with other sensor nodes and possibly a gateway. These mechanisms may be wired or wireless. The communications module houses all these mechanisms. Detailed modelling of the communications mechanisms in a sensor node is beyond the scope of this part of ISO/IEC 29182. Such modelling should include at least the physical and data link layers in the Open Systems Interconnection (OSI) model and possibly the network and transport layers. Appropriate standards should be cited to characterize the protocol layers.

6.1.5 Processor

A sensor node, especially if it uses multiple sensors, typically has a processor that may be used for pre-processing raw sensor data. This could include data aggregation, feature extraction, data fusion, and even collaborative processing of data captured not at just the node under discussion but also at other nodes communicating with it. The communications module and functionalities such as security services at a sensor node require computational capabilities. In general, the term "smart sensor" – which is commonly used – implies that there is a good bit of processing capability at the sensor node. From a functional point of view, many entities require processing and computational capabilities. These include Basic Functions Layer (BFL), Service Layer (SL), Application Layer (AL), and Cross-Layer Management (CLM), which are described later under Functional Entities.

A detailed architectural model for the processor is not needed for characterizing sensor node capabilities. Perhaps all that is needed is a simple characterization using FLOPS (FLoating OPerations per Second) and the number of processors (dual, quad, etc.) to specify the computational power of the node.

6.1.6 Memory

The storage capabilities of a sensor node can be characterized with two numbers: the size of its hard drive (slower access memory) and the size of its faster electronic memory.

6.1.7 Power supply

Sensor nodes are typically battery-powered. Therefore, the battery voltage and its longevity, characterized in terms of milliampere-hours, need to be specified. Some sensor nodes use a sleep mechanism that allows them to go to "sleep" most of the time and "wake up" only occasionally to do what a sensor node does – data acquisition, processing, and reporting through communications with other entities. In such cases it might be useful to characterize the recharging capabilities of the batteries, because batteries recharge when they are not in use.

A battery-operated sensor node may also use some form of energy harvesting, which needs to be characterized in order to estimate how long the sensor node may last.

In certain applications, sensor nodes are powered with line electricity. In such cases, it is useful to specify the power consumption of the sensor node.

Power supplies – whether they are batteries or Alternating Current (AC) adaptors – are typically bulky and constitute a major fraction of the weight and size of a sensor node. Therefore, it is useful to specify the size and weight of the power supply.

6.2 Gateways

Gateways facilitate communications between a sensor network and another network or another sensor network. If present in the overall architecture, gateways reside in physical proximity of sensor nodes. A gateway employs one protocol for communicating with a sensor network and another for communicating with the other network, whether it is another sensor network or a backbone network. In the latter case, the communications are either direct or through an access network. Just as in the case of the communications module used in sensor nodes (cf. Subclause 6.1.4), a full specification and modelling of the communications protocols used by a gateway is beyond the scope of this part of ISO/IEC 29182. Wherever appropriate, other standards should be cited.

6.3 Other networks

6.3.1 Overview

"Other networks" refers to the networks in the SNRA other than sensor networks. These are the networks that make it possible for a sensor network to communicate with its users through service providers, particularly when the users and service providers are not physically co-located with the sensor network. Other networks include access networks and the backbone network that are described next. Note that there is no need for "other networks" in a stand-alone sensor network, as shown in Figure 1 of ISO/IEC 29182-1[1].

6.3.2 Access networks

An access network provides connectivity between the backbone network and a gateway in the SNRA. Examples of access networks include a WiFi network, a cellular telephony network (such as 3G/4G (3rd Generation/4th Generation) Wireless), and simply an Ethernet in the case of a gateway that is hard-wired to the backbone network. The modelling of an access network is beyond the scope of this part of ISO/IEC 29182. Wherever appropriate, other standards should be cited.

6.3.3 Backbone network

The most obvious backbone network is the Internet. Another example would be an intranet, if the data from the sensors is going to be consumed "locally" and not to be accessed by other networks. In general, a backbone network provides connectivity among a large number of possibly geographically dispersed communicating entities. It is typically wired, even though wireless backbone networks have also been proposed. The modelling of the backbone network is beyond the scope of this part of ISO/IEC 29182. Wherever appropriate, other standards should be cited.

6.4 Service providers

These are entities that interact with one or more sensor networks and provide some basic services to the users of these sensor networks and more specifically to the sensor network applications running on the user machines. For example, sensor network applications such as national air traffic control and battlefield command and control rely on some basic services related to weather and climate monitoring. These applications may get weather information from service providers providing such information and forecasting services.

6.5 Users

These are the entities that ultimately consume the high level information provided by sensor networks. Sensor network applications, such as environmental monitoring and battlefield command and control, run on the user machines. As pointed out in Subclause 6.4, these applications may rely on certain basic

services provided by service providers. The user may have capabilities for visualizing the information produced by sensor network applications. It should also be pointed out that in case of simpler sensor networks, the applications may run on lower layer entities and even on sensor nodes.

7 Functional entities

7.1 Sensor node hardware layer

7.1.1 Overview

The sensor node hardware layer is the collection of functionalities dealing with sensing, actuation, and power sources in a sensor node. Obviously, sensing is the main function of a sensor node, but it may also have actuation capabilities. Sensing and actuation take place strictly in sensor nodes, but other physical entities require power sources also. While those entities often have access to line electricity, the issue of power generation in sensor nodes deserves special attention because they may have to live off batteries for a long time or harvest energy.

7.1.2 Data acquisition

This is the fundamental function of sensors. Each sensor measures some physical attribute of the environment in which it is located and converts those measurements into digital data. Some details on the operation of a sensor have been given in Subclause 6.1.2. Data acquisition is the collective task of observing and measuring the environment and producing digital data based on the measurements.

7.1.3 Actuation

In the context of a sensor network, this is the process through which a user may affect the physical world that the sensors observe and measure. Not all sensor networks are equipped with actuators. There are some sensor networks that simply observe and measure some attributes of the physical world without trying to affect it. One may regard a sensor network that employs actuators as a closed loop control system. Each actuator receives control commands/signals from higher functional layers of the sensor network and causes some object to move. The notion of how to affect a possibly "large" physical world through use of a number of actuators in a way that meets the user's goals and expectations is a challenging problem and in a sense the dual of the sensor data fusion problem.

7.1.4 Power generation / energy harvesting

Not all sensor networks harvest energy. A sensor network with battery-operated sensor nodes will have a limited lifetime. It will die and stop functioning when a sufficient number of its nodes run out of battery. One way of mitigating this problem is to periodically replace the batteries. However, this is not convenient and it can be costly in case of a large sensor network. One other way is through use of energy harvesting, where a sensor node uses some means of extracting energy from the environment in which it is located. The most prominent example of energy harvesting is through use of solar cells. Another possibility is through use of wind turbines.

7.2 Basic functions layer

7.2.1 Overview

Aside from data communications and data storage, the functional entities described under this entity are strictly associated sensor nodes. Sensor nodes do data communications and data storage, but so do other physical entities in the SNRA. Beyond sensing, actuation, and power generation, there are some basic tasks that sensor nodes need to do. Those functionalities are grouped under "basic functions layer" and are described next.

7.2.2 Data processing

A sensor node may employ various algorithms to process the raw data acquired by its sensors. Averaging and filtering (linear or nonlinear) for removing additive or speckle noise are examples of these algorithms. Data aggregation and data compression are other examples motivated by the fact that the communication bandwidth – particularly in the case of wireless sensor nodes – is often a scarce commodity. Therefore, it makes sense to process the raw data and reduce the volume of the data that has to be communicated to neighbouring sensor nodes or a centralized processing entity that receives data from all nodes in the network. The notion of sufficient statistics is well understood in the statistics literature. The idea is to process the raw data and extract from it a much smaller data set, called the processed data, which captures the "essence" of the raw data. Mathematically speaking, the optimum decision based on the sufficient statistics would be as good as the optimum decision based on the raw data in its entirety. One example that illustrates the use of sufficient statistics is detecting presence of hostile forces in an area under military surveillance using many sensors. One can either transmit the raw data from all sensors to a processing entity or transmit the sufficient statistics. If the optimum decision rule regarding the presence of hostile forces based on the smaller data set has the same probabilities of detection and false alarm, or more generally the same Receiver Operating Curve (ROC), as the optimum decision rule based on the raw data, then the smaller data set is called a sufficient statistics. Along the same lines, feature extraction is another type of processing that attempts to reduce the volume of the data that has to be sent by the sensor nodes to other entities. A good example for feature extraction is in the context of image and video data, where some important features of an image or video clip, such as presence of certain objects, is captured and transmitted in lieu of sending every single pixel in the image or all frames in the video clip.

Another aspect of data processing is data presentation and format. There has to be some agreed upon way of interpreting the data exchanged by various entities. For example, when raw temperature data are sent by one sensor node to another node or the central processing entity, there has to be a header that specifies the units for temperature, e.g. Celsius vs. Fahrenheit, and perhaps the resolution of the sensor. This requires presence of data presentation functions that add extra information to basic sensor measurements in order to put the data in context.

7.2.3 Data communications

Data communications takes place between various physical entities in a sensor network as suggested by Figures 1-3 in ISO/IEC 29182-1. The communications can be through a wire or an air interface (wireless) and various protocols can be used for communications between various entities. The primary mode of information exchange in the backbone is wired communications. In addition, the sensor nodes use wired communications in certain deployments, for example sensors deployed in some buildings. Wireless communications offer many advantages, among which are support for mobility and ease of deployment. However, one has to deal with Radio Frequency (RF) interference issues, and it is physically easier to eavesdrop on wireless transmissions.

The Internet uses the Internet Protocol (IP) at the network layer, but communications between some entities in a sensor network – e.g. inter-sensor-node communications – may not be IP-based due to the relatively large overhead of the IP. IEEE 802.15.4a, ZigBee, and IEEE 802.11 are some of the wireless communications standards commonly used for inter-sensor-node communications. These standards deal with the Physical (PHY) and Medium Access Control (MAC) layers of the protocol stack. Likewise, there are a variety of standards that could be used by an access network. In the case of a wired access network, Ethernet is the dominant Local Area Network (LAN) standard. There is a wide choice when it comes to wireless communications standards. Examples include Bluetooth, IEEE 802.11 Wireless Local Area Network (WLAN), and cellular telephony standards, such as 3G wireless standards (e.g. Universal Mobile Telecommunications System (UMTS) and High Speed Packet Access (HSPA)) and 4G wireless standards (e.g. Long Term Evolution (LTE) and Worldwide Interoperability for Microwave Access (WiMAX)).

Naturally, when it comes to data communications, a designer needs to specify other protocols used at the higher layers of the OSI model, in particular network and transport layers. It is not the intent of this part of ISO/IEC 29182 to address all aspects of data communications, as there are many other standards that deal with those aspects.

7.2.4 Data storage

A sensor node may have storage for storing its raw/processed sensor data for some period of time. This is useful in looking for "trends" in the acquired data or more generally for use by the data processing algorithms that typically need more than just the latest sensor measurements.

There is data storage in other entities of a sensor network, e.g. historical database of events maintained by service providers. This aspect is addressed elsewhere in this part of ISO/IEC 29182.

7.2.5 Hardware drivers

Device drivers are commonly used in a computer system to enable communications between the computer and the devices connected to it. Similarly, they can be found in sensor nodes so that the node can operate sensors and actuators. They work in conjunction with the processor at the sensor node.

7.2.6 Sensor/actuator identification

The presence and types of sensors and actuators at a sensor node can be specified via the Transducer Electronic Data Sheets (TEDS) defined in the ISO/IEC/IEEE 21450 standard[3] and the ISO/IEC/IEEE 21451 series of standards[4]. A TEDS contains information on sensor or actuator identification (ID), physical units (such as degrees Celsius in case of temperature), measurement range, calibration, and location (which might be updated by a self-localization capability such as Global Positioning System (GPS) in case of a mobile sensor / actuator), user-specified information, manufacturing-related information, and more. The TEDS is a means of self-identification and self-description of sensors and actuators, as well as self-configuration of sensor systems. It simplifies field installation, upgrade, and maintenance of sensors and actuators in systems, thus enabling sensor "plug and play" in instruments and networks.

7.3 Service layer

7.3.1 Overview

The functionalities described under Subclause 7.3 are distributed across many physical entities in the sensor network, including sensor nodes, service providers, and users. In general, a variety of services are available in a sensor network that are used by various applications running on the sensor network.

7.3.2 Common services

7.3.2.1 Overview

The services described below are common in the sense that they support a variety of sensor node applications.

7.3.2.2 Communications support functions

Communications and networking require some support functions across various layers of the protocol stack. For example, at the PHY layer there is need for error correction coding. At the network layer there is routing, which can be computationally intensive in ad hoc and mesh sensor networks. Network formation is another communications support function at the network layer. This is an important functionality in ad hoc and mesh sensor networks, through which a sensor node discovers who its neighbours are that it can communicate with.

7.3.2.3 Self-localization

More often than not, sensor data are more useful and meaningful if it is tagged with the time and location at which it was acquired. Time synchronization is addressed in Subclause 7.3.2.7. As for localization, some expensive sensor nodes may have a GPS receiver embedded in them. In that case, the sensor node would know its location and the time through reception of messages from GPS satellites. More generally,

the same functionalities are provided by the Global Navigation Satellite System (GNSS). GNSS, however, would work only if the GNSS receiver is in Line-Of-Site (LOS) of at least 4 GNSS satellites. Therefore, GNSS would work and provide localization capability if the sensor node is located outdoors, excluding urban canyons, and more generally not in GPS-/GNSS-denied environments. Another drawback of GPS/GNSS is its high power consumption, which makes it unsuitable for most sensor nodes that are expected to run on battery power for a long time.

Fortunately, there are other methods by which sensor nodes can self-localize. This is an active area of research and development, but suffice it to say that there are RF methods (e.g. Time Of Arrival (TOA), Angle Of Arrival (AOA), Received Signal Strength (RSS) based methods, and Radio Frequency IDentification (RFID)), Inertial Measurement Units (IMUs), magnetometers, altimeters, Doppler radar, and many other sensors that can be used for localization, particularly in indoor environments. The task of combining various sensor outputs in order to arrive at an accurate estimate of the sensor node's location is a data fusion problem, which once again is the subject of active research.

There are also cooperative localization methods that solve for the locations of *all* sensor nodes in the network jointly as opposed to solving for each node's location independently. This results in more accurate location solutions, because one can take advantage of simple geometric facts such as the triangle inequality. Once again, there are both centralized and distributed cooperative localization algorithms.

7.3.2.4 Service/resource discovery

Service Discovery Protocols (SDPs) are used to determine what sensor network services are available. For example, one may wish to determine if there is an unoccupied parking spot locator service when one enters a parking lot/structure. This is just one example of a wide array of services called Location-Based Services (LBS). Jini, Service Location Protocol (SLP), and Simple Service Discovery Protocol (SSDP) as used in Universal Plug and Play (UPnP) are all but a few examples of SDPs.

7.3.2.5 Data management

Data management deals with sharing of sensor data for use by various sensor network services, synchronization of such data, and certain level of data fusion and processing. It also deals with database management issues related to archiving sensor network data for future use as historical data.

ISO/IEC JTC 1/SC 32 "Data management and interchange" deals with data management in distributed information systems. It covers areas such as (a) reference models and frameworks for the coordination of existing and emerging standards, (b) definitions of data domains, data types, data structures, and their associated semantics, and (c) languages, services, and protocols for persistent storage, concurrent access, concurrent update and interchange of data; methods, languages, services and protocols to structure, organize and register metadata and other information resources associated with sharing and interoperability, including electronic commerce. In the context of sensor networks, it deals with storage, discard, and mining of information from sensors.

Specifically, ISO/IEC 9075 is a family of standards on database language SQL, the lingua franca for storing/retrieving/manipulating data in Relational Database Management Systems (RDBMS). ISO/IEC 9075-1[5], Framework, describes the conceptual framework used in other parts of ISO/IEC 9075 to specify the grammar of SQL and the result of processing statements in that language by an SQL-implementation. It also defines terms and notation used in the other parts of ISO/IEC 9075. ISO/IEC 9075-2[6], Foundation, specifies the structure of SQL-statements and the effects of executing them. It has the bulk of what most people regard as SQL. ISO/IEC 13249 series are SQL/Multi-Media standards - i.e. standards that support the manipulation of specialized content (such as full-text, spatial, still image,...) in SQL databases.

ISO/IEC JTC 1/SC 32 has also developed standards on metadata. ISO/IEC 11179 – Metadata registries – is a 6-part International Standard on the semantics and representation of data and a registry to manage them. ISO/IEC 19763 – Metamodel framework for interoperability – is a 12-part International Standard on managing and linking similar models throughout information management. ISO/IEC 20944 – Metadata registry interoperability and bindings – is a 5-part International Standard on bindings (codings, Application Programming Interfaces (APIs), and protocols) from programming languages to registries.

7.3.2.6 Code management

Code management is the process of managing the changes to documents, programs, and other information stored as computer files in an information system. This is also called version control, revision control, or Source Configuration Management (SCM). In the context of sensor networks, the programs may be code for signal processing algorithms that run on sensor data. These algorithms may need access to historical sensor data, topographic information, and other types of information that need to be periodically updated. As new code or data files become available, they need to be pushed to *all* entities that run that code or access the information contained in the data files. Use of different versions of a piece of code by various entities may lead to unintended consequences in the sensor network and the services it provides.

7.3.2.7 Time synchronization

There are a number of functionalities in a sensor network that need a time service. It has been already pointed out that sensor data are often more useful when it is time-stamped. That would require synchronization among sensor nodes. Synchronization may also be needed for data communications when coherent modulation schemes are used at the PHY layer of the protocol stack. As it has been stated elsewhere in this part of ISO/IEC 29182, one way of obtaining time service is through use of GPS/GNSS.

IEC 61588:2009(E) or IEEE Std 1588-2008[Z] is a standard for Precision Time Protocol (PTP), which can be used for precise time synchronization of sensor nodes.

7.3.2.8 Group management / clustering

In certain cases it is best to partition the nodes in a sensor network into several groups, also called clusters. For example, this would be a sensible thing to do when the sensor network is made up of many sensor nodes distributed over a large geographic area and there are distinct groups of nodes whose members are in close proximity of each other. Since there is typically a high degree of correlation in the raw data collected by the sensors in such nodes, the members of each such group may elect one group member as cluster-head. The job of the cluster-head is to collect raw data from the nodes in that group/cluster, process that data, and communicate a concise summary of that data to other cluster-heads or other entities in the sensor network. The summary may include some statistics (mean, variance, median, x-percentile, minimum, maximum) of the data collected by the cluster. This reduces the data communications burden. Typically, communications within a cluster in a wireless sensor network are simple and do not require routing as the cluster-head is capable of directly communicating with all the nodes in the cluster. This sets a constraint on the size of a cluster. Its radius should be no larger than the radio communication range of the sensor nodes. The cluster-heads from various clusters then communicate with each other and with other entities in the network – such as service providers and users – and exchange information. These exchanges may go through gateways, access networks, or the backbone network. The range for inter-cluster communications is larger than that for intra-cluster communications. The cluster-head may be identical to all other nodes in the cluster or it may be a special, more powerful node. In the former case, the cluster-heads need to use higher transmit powers in order to communicate with other cluster-heads or other sensor network entities. This is done through transmit power control, which is also used for topology control discussed in Subclause 7.5.5. In the latter case, the cluster-head may have a more powerful radio that allows it to communicate over larger distances than ordinary sensor nodes.

Group management / clustering deals with how clusters are formed and maintained. There are different objective functions that can be optimized for clustering. Aside from considerations related to which sensor nodes have correlated data and hence are candidates for being grouped together, there may be energy consumption considerations for maximizing the lifetime of the sensor network. Once some sensor nodes or cluster-heads stop working for any reasons (running out of battery or getting out of range due to mobility of sensor nodes), then some steps must be taken to repair the clusters and maybe form new ones. All this is handled by the group management / clustering service.

7.3.3 Domain specific services

Domain specific services support the development of applications for specific market segments or application domains. For example, the security needs of various applications may be different depending on legal, cultural, organizational, and ethical issues related to the application. This would affect how data and information related to that application needs to be handled and protected. There may also be specific types of data processing and data presentation required by a specific application domain. Two applications domains related to logistics and security have been shown in Figure 2 as examples. This is by no means meant to be an exhaustive listing. The logistics domain deals with keeping track of where certain objects are, a classification of objects, and their quantities, such as in a supply chain management system. One important application in the security domain is intrusion detection, as in a multi-sensor burglar alarm system.

7.4 Application layer

7.4.1 Overview

The functionalities described under Subclause 7.4 are distributed across many physical entities in the sensor network, including sensor nodes, service providers, and users. Each sensor network application relies on a number of services provided by the service layer (cf. Subclause 7.3).

7.4.2 Applications

A sensor network application performs value-adding operations on raw or processed sensor data according to the user needs and makes available the ultimate output of these operations as a service to the user. A number of sensor network applications and services have been envisaged, and many more are yet to come once more networked sensors are deployed all around us. Examples include applications for service domains such as healthcare, Intelligent Transportation Systems (ITS), environmental monitoring, military, logistics and supply chain management, and energy and utilities, such as smart grid systems.

The model for a sensor network application needs to specify the types of sensors available to the application, the precision of each sensor type, the signal processing algorithms that take whatever raw or processed sensor data available to the application – taking into account the possibility of packet loss in the communications paths – and convert that to the high level information that the user needs according to the user requirements.

7.4.3 Software agent

A software agent is a piece of code that acts on behalf of a sensor network application or user with some degree of autonomy. It typically does its work based on a high-level description of what needs to be done by the application or user and without detailed instructions. There are various types of software agents, including intelligent agents that exhibit some learning and reasoning aspects, autonomous agents that are capable of modifying the way in which they achieve their objectives, distributed agents that are executed on physically distinct machines, and mobile agents that can relocate their execution onto different processors.

7.4.4 Rules engine

A rules engine is, at its core, a software system or mechanism for executing one or more “business rules” in a runtime production environment. Business rules might come from legal regulations, company policy, or other sources. They are simple business-oriented statements that encode business decisions of some kind, often phrased very simply in an if/then conditional form. For example, in the context of a sensor network applications for the utilities domain, there will be business rules that define how much a customer will be charged for electricity based on conditions defined on a variety of sensors reflecting a supply and demand model. Another example is in the context of data privacy. There may be a business rule that requires encrypting identities of mobile users in certain LBS. In other words, LBS applications

may have access to the locations of mobile users in certain venue, such as boots at a trade exhibition, but not the identities of the users.

7.4.5 Collaborative information processing

A sensor network may operate in a centralized or distributed manner. In the former case, the data from all sensor nodes, whether raw or processed, is sent to a central entity that processes the data and infers from it some high level information about the environment being observed by the sensor network. This could be in the form of a decision about some underlying phenomenon, e.g. presence of hostile forces or lack thereof as described in Subclause 7.2.2, which is called a detection or hypothesis testing problem, or estimating some continuous random object, such as the random field of water temperature in some region of an ocean, which is an estimation problem. This process is called data fusion regardless of whether all the sensors are of the same type or sensors with different modalities – such as acoustic, seismic, and optical – are involved. In the centralized case, data fusion takes place in the central entity. In the distributed case, there is no central entity that receives and processes all the data. Instead, the sensor nodes exchange information with their neighbours, do some data fusion, and this cycle repeats a number of times until decisions/estimates converge. This is called collaborative information processing. In this case every sensor node may have that final decision/estimate or at least some high level information about its surroundings based on not just its own sensor measurements but also data from other sensor nodes.

It is simpler to design a sensor network that operates in a centralized fashion, but the data communications burden is higher and the network would have a single point of failure. That is, if the central processing entity fails, then the whole network fails. It is harder to design a sensor network that operates in a distributed manner, because it is not easy to decide what information should be exchanged with neighbours and how to process the data so that the decisions/estimates would be as good as with centralized operation or at least would be useful. However, a distributed sensor network degrades gracefully if some of the sensor nodes or other entities in the network fail. There are also hybrid designs where all data produced by sensor nodes within a geographic area or cluster is sent to a super node or cluster-head and "fused" there and then the cluster-heads further process their data in a distributed manner.

Even though the discussion of centralized vs. distributed data processing has appeared under "Application Layer", the functionalities and operations described above need not take place exclusively in this layer. Some of this may take place in the Basic Functions Layer or Service Layer.

7.5 Cross-layer management

7.5.1 Overview

There are many functionalities in a sensor network that are spread over various functional layers shown in Figure 2. Cross-layer management deals with managing such functionalities.

7.5.2 Device management

This entity manages the devices comprising a sensor network. It applies primarily to sensor nodes and gateways. Perhaps the most important resource that has to be managed in a battery-operated sensor node is the battery power. The sensor node may have a sleep mechanism that forces it to go to sleep most of the time and "wake up" occasionally to sense the environment, process the raw sensor data, and transmit it to some other entity, per discussions under data processing and collaborative information processing subclauses. Another alternative is to have a less power-hungry sensor stay on all the time to detect onset of events of interest and have it wake up more capable sensors to more precisely sense the environment and "measure" the event. Naturally, these sensors would go to sleep after the event is over, while the less capable sensor stays on. The same concept may be used in conjunction with the radio used at a wireless sensor node, because radios consume nearly as much power in standby mode as in transmit and receive modes. There may be a simple radio that continuously monitors the airwaves and looks for "request to send" messages from other devices that may wish to communicate with the sensor node. Upon detection of such a message, the simple radio may send a "clear to send" message and wake up a more powerful radio that would actually receive the message from the transmitting device.

Aside from managing sleep schedules, the device manager at a sensor node has the capability to adjust the rate at which sensors sense the environment. It also manages other entities in a sensor node, such as data processing. Sometimes it is cheaper from an energy consumption point of view to process the raw sensor data at the node and then transmit the processed data to other entities. At other times, it might be the other way around. It all depends on the amount of energy required by each alternative. Yet another aspect of device management is radio-transmit power control, just as it is done in cell phones. A sensor node communicating with another close by sensor node or other device should reduce its transmit power to a level barely enough for the other device to hear the transmissions. This reduces power consumption at the transmitting sensor node as well as the RF interference to other radio communications. However, this aspect, if not covered by the communications support functions (cf. Subclause 7.3.2.2), is one of many functionalities of the networking protocol stack used by the sensor network, which is beyond the scope of this part of ISO/IEC 29182.

In case of mobile sensor nodes, the device manager controls the movements of the node. Some of these movements are autonomous in the sense that the node decides on its own where to move to. Other movements are directed by other sensor network entities, such as the resource management entity (cf. Subclause 7.5.3).

Gateways need management also. A gateway acts as a translator. It is capable of communicating over a number of networking protocols with sensor nodes and over another set of protocols with access networks or the backbone network. A gateway should be capable of establishing communications with a sensor node over any of the protocols it supports for such communications. It may not a priori know which protocol the sensor node uses, the sensor node may move beyond the communication range of the gateway, and new sensor nodes, possibly from a different sensor network, may move in the vicinity of the gateway and wish to use it to communicate with “other networks” (cf. Subclause 7.3). A gateway should be capable of handling all these situations.

7.5.3 Resource management

Resource management deals with managing a multitude of physical and functional entities across the sensor network and other supporting entities, not just one device as in the case of device management. It takes into account interactions among such entities. For example, the resource manager for the entire senor network may decide to reduce sensing activity in certain geographic area covered by the sensor network and increase activity in others. This may be prompted as a result of examining the high level information coming from sensor data fusion and collaborative information processing that suggests certain areas are of less/more interest. Alternatively, it might be due to congestion in data communications pipelines, whether wired or wireless, or degradation of radio channels that makes it imperative to control the amount of data that certain sensor nodes generate and transmit. Then those sensor nodes need to be instructed to adjust the rate at which they sense the environment or transmit data. As for the latter, in the simplest case a sensor node may be asked to aggregate many measurements and send one value instead of all raw sensor measurements.

Another aspect is to extend the life time of the sensor network as much as possible. Then sensor nodes with critically low battery power need to be instructed to reduce their activities or even go into hibernation until somehow their batteries are recharged as a result of inherent recharging in the battery or some means of energy harvesting. This may make it necessary to move some mobile sensors to areas where sensor nodes are about to die. When mobile sensor nodes are available, the resource manager has the luxury of instructing nodes to move to areas not previously under surveillance by the sensor network. Any move decision may also be affected by sensor measurements in a data-driven fashion.

In short, the resource management entity has an overall view of the entire sensor network and often performs cross-layer optimizations to arrive at resource allocation and management decisions.

7.5.4 Service management

This entity deals with service registration, service description, service analysis, and it maintains the service processing queue.

7.5.5 Network management

This entity deals with managing communications within the sensor network and between the sensor network and service providers and users through gateways, access networks, and the backbone network. This is not about managing the access networks or the backbone network, which have their own management entities. However, information about the status of those networks is taken into account in managing the sensor network. Therefore, the focus of this entity is on managing the sensor network.

Network management is a broad area and there are many aspects of a network that need to be managed. Examples include topology management, routing table management, performance management, and configuration management. Topology control in a sensor network dictates which sensor nodes a given sensor node communicates with, even though there may be other nodes that can "hear" the given sensor node. This is particularly important in wireless ad hoc and mesh sensor networks, because it reduces the complexity of routing messages. One approach to topology control is to build a backbone over the sensor network. The backbone typically has a tree structure, which implies that the route between any two nodes is unique and hence routing becomes trivial. However, with a tree structure the network becomes disjoint if any link fails. Therefore, it is worthwhile to consider other network topologies and introduce some redundancy in choice of routes between any pair of nodes.

Routing becomes more interesting in wireless ad hoc and mesh sensor networks, as the routes often break and other routes need to be found. There is a large body of literature on this topic and many, many routing protocols have been proposed for ad hoc and mesh networks, with or without mobility.

Performance management is more interesting and challenging in a sensor network than in a communication network, because the sensor network has not only the communications aspects, but also the signal processing and inference issues. Basically, communications is just a means for the sensor network to do what it is supposed to do. The first task in managing the performance of a sensor network is to define appropriate performance metrics and specify how to measure them. This is closely related to system monitoring (cf. Subclause 7.5.11). Once problems with and degradations in sensor network performance are detected, corrective action is taken to the extent possible to remedy those problems.

Configuration management is a more general case of SCM (cf. Subclause 7.3.2.6). It deals with modifying the way sensor nodes and other entities such as gateways behave to meet certain networking objectives.

7.5.6 Security management

Security is of paramount importance in sensor networks just as it is in communication networks. It becomes even more crucial in a sensor network with actuation capabilities just as in any industrial control system. Any lapse in security that yields control of actuators to unauthorized individuals, possibly with malice intent, can have devastating consequences.

Security management deals with authentication, authorization, availability, and even routing security. Most of these concepts are well understood in the Information Technology (IT) domain and will not be explained in any depth in this part of ISO/IEC 29182, except for authentication. It is vital to confirm the authenticity of the sensor nodes that are allowed to connect to a sensor network and ensure that only authorized users are allowed to access the sensors and the data they generate via authorized service providers. User authentication is the process of associating an individual with the user's unique identity. Passwords or more complex forms of multi-factor user authentication are commonly used to control user access. A sensor node should be uniquely identified before it can be registered into the sensor network.

Note that sensor networks are, more often than not, resource-constrained. Therefore, security solutions need to be "lite". They should be easy to deploy, require minimal manual operation, and be insofar as possible based on existing security standards. ISO/IEC 27001:2005[8] specifies the requirements for establishing, implementing, operating, monitoring, reviewing, maintaining and improving a documented Information Security Management System within the context of an organization's overall business risks. It specifies requirements for the implementation of security controls customized to the needs of individual organizations or parts thereof. ISO/IEC 27002:2005[9] establishes guidelines and general principles for initiating, implementing, maintaining, and improving information security management in an organization. The objectives outlined provide general guidance on the commonly accepted goals of information security management.

7.5.7 Privacy management

Certain data in a sensor network might be private. There are many examples of such data in military domains. Even in commercial applications, the privacy of certain data needs to be protected. For example, with the proliferation of smartphones that are equipped with a variety of sensors, it is plausible to use these phones for environmental monitoring. The sensors on the phones can measure various physical attributes and send them to a central authority. Neither the data they collect nor the locations at which the data has been collected are private. The identity of the owner of the smartphone, however, is private, because otherwise someone can track the owner's movements. More generally, there are other types of data that falls under the category of Personally Identifiable Information (PII) that needs to remain private.

This entity is about encryption, key management, and privacy protection in a sensor network. In some cases the encryption needs to be end-to-end. In other instances, the privacy of data needs to be protected in certain entities in the overall architecture.

Just as with security management (cf. Subclause 7.5.6), methods for protecting privacy of data in sensor networks need to be "lite". ISO/IEC 29192[10] is a multi-part International Standard that specifies lightweight cryptography for the purposes of data confidentiality, authentication, identification, non-repudiation, and key exchange. Lightweight cryptography is suitable in particular for constrained environments. The constraints normally encountered can be chip area, energy consumption, program code size and RAM size, communication bandwidth, and execution time. As a matter of fact, all of these constraints apply to sensors deployed in forests, deserts, etc. for environmental monitoring. The purpose of ISO/IEC 29192 is to specify standardized mechanisms which are suitable for lightweight cryptographic applications, including RFID tags, smart cards (e.g. contactless applications), secure batteries, health-care systems (e.g. Body Area Networks), sensor networks, etc.

ISO/IEC 29100[11] provides a high-level framework for the protection of PII within Information and Communication Technology (ICT) systems. It is general in nature and places organizational, technical, and procedural aspects in an overall privacy framework intended to help organizations define their privacy safeguarding requirements related to PII within an ICT environment by: (i) specifying a common privacy terminology, (ii) defining the actors and their roles in processing PII, (iii) describing privacy safeguarding requirements, and (iv) referencing known privacy principles.

7.5.8 Safety management

It is important to ensure that a sensor network does not endanger the safety of people or assets. It is rather obvious that actuators are a safety concern and can injure people or cause significant damage to assets (cf. Subclause 7.5.6). A sensor network is a safety concern in terms of the possibility of missing detection of important events or causing false alarms. If the suite of sensors monitoring the physiological health of a patient or a firefighter entering a building on fire miss the onset of a cardiac arrest, then obviously this is a cause for concern. On the other hand, a false alarm may cause initiation of some response that might have safety implications. Safety management is quite important in industrial control systems.

7.5.9 Business management

This entity deals with the business aspects of a sensor network, such as accounting. It keeps track of which sensor network services are provided to each user and the details for each instance of usage, so that the user is appropriately billed. Naturally, this involves keeping electronic records as well.

7.5.10 QoS management

The notion of Quality of Service (QoS) is well understood in image and video communications. In image communications, the user may ask for certain pixel resolution and depth or certain quality factor used by the image coding scheme such as the Joint Photographic Experts Group (JPEG) coding method. In video communications, in addition to pixel resolution and depth and coded video quality, the user may ask for certain frame rate and end-to-end transmission delay. Alternatively, the user may demand maximization/minimization of certain QoS measure. For example, the user may ask for the best possible image quality, irrespective of how long it takes to get that image.

Demanding certain level of QoS is one thing, provision of that level of QoS is another! The latter is straightforward in case of point-to-point communications over a single link and a well-known communication channel, but it becomes more complicated when the communications are over a network. In the latter case, the QoS requirement has to be mapped to requirements on various layers of the protocol stack for all the nodes that are on the communication path from the source to the destination. There will be requirements on the PHY layer (such as transmission rate and bit-error-rate), MAC layer (how fast a node gets to use the channel), transport layer (e.g. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP)), routing layer, and application layer (quality of the image/video signal at the point of acquisition and how aggressively it is compressed). This is hard to do and guarantee over a wired backbone network. It becomes harder when some over the air (wireless) links are on the path, and it becomes very challenging in the cases of wireless mesh and mobile ad hoc networks. That's why there are many problems in QoS provision over a network that are still open and this remains an active area of research and development.

Sensor networks may use image/video cameras, but they may also employ a wide array of other types of sensors, such as acoustic, seismic, temperature, humidity, light, infrared, and gas sensors. Defining the quality of measurements made by these sensors at the point of data acquisition is straightforward. One would then have to deal with the problems associated with communicating over a network that were described earlier. However, in case of sensor networks, QoS is defined at a higher level by the sensor network service and application layers. For example, consider a situation where many types of sensors are deployed along the border between two countries for a border monitoring application. The user does not care what kind of data is received from the video surveillance cameras, acoustic sensors, and seismic sensors. What matters to the user is to have guarantees that the probability of detecting a border breach by an intruder is above certain level, while the probability of false alarm is kept at bay and below some other level. So, QoS does not depend on data acquisition and communications only, but also on how data are processed. Therefore, there is a need for an extra step to map the QoS requirements at the sensor network application layer to requirements at the service layer and then requirements on what data quality is needed from each sensor by the processing algorithms such as data processing and collaborative information processing blocks. This is easier to do in the case of centralized processing of data than when a distributed algorithm is used to process the sensor measurements. Yet another aspect is the coverage area of a sensor network. In case of mobile sensors, the user may wish to change the coverage area of the sensor network and focus on some other region for monitoring. This may also be regarded as a QoS measure.

In short, the notion of QoS in sensor networks is more involved than its counterpart in communications and networking.

7.5.11 System monitoring

System monitoring is about real-time tracking of how sensor network entities are functioning. Its purpose is to detect failures in the network. This requires provision of some special sensors that detect failures in entities. Another way of detecting failures is through inference. For example, if a temperature sensor reports readings that are unusual at a given time for a certain location or is stuck at some value over an extended period of time, then the monitoring agent would conclude that the sensor has probably failed. Corrective actions are taken in response to component failures to the extent possible.

Bibliography

[1] ISO/IEC 29182-1, *Information technology — Sensor networks: Sensor Network Reference Architecture (SNRA) — Part 1: General overview and requirements*

[2] ISO/IEC 29182-3, *Information technology — Sensor networks: Sensor Network Reference Architecture (SNRA) — Part 3: Reference architecture views*

[3] ISO/IEC/IEEE 21450, *Information technology – Smart transducer interface for sensors and actuators – Common functions, communication protocols, and Transducer Electronic Data Sheet (TEDS) formats* [IEEE 1451.0]

[4] ISO/IEC/IEEE 21451 (all parts), *Information technology – Smart transducer interface for sensors and actuators* [IEEE 1451.x]

[5] ISO/IEC 9075-1:2011, *Information technology — Database languages — SQL — Part 1: Framework (SQL/Framework)*

[6] ISO/IEC 9075-2:2011, *Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation)*

[7] IEC 61588:2009(E), *Precision clock synchronization protocol for networked measurement and control systems* [IEEE Std 1588-2008]

[8] ISO/IEC 27001:2005, *Information technology — Security techniques — Information security management systems — Requirements*

[9] ISO/IEC 27002:2005, *Information technology — Security techniques — Code of practice for information security management*

[10] ISO/IEC 29192 (all parts), *Information technology – Security techniques – Lightweight cryptography*

[11] ISO/IEC 29100:2011, *Information technology — Security techniques — Privacy framework*

ICS 35.110
Price based on 22 pages