Computer Science • 13 (4) 2012 http://dx.doi.org/10.7494/csci.2012.13.4.101

Marek Konieczny

ENRICHING WSN ENVIRONMENT WITH CONTEXT INFORMATION

Abstract *Currently the Internet of Things environment suffer from a variety of unique problems such as low throughput, inadequate support for mobility and not-compatible (and often close) application environments. However, characteristic of physical environment often provides opportunities to address these problems. This paper explores concepts of enriching the WNS environment with context-based information. By context we consider all data available in sensors environment, not only data processed by them. Currently this data are not utilized by sensors, but we think that they can improve overall functionality of the sensor network. We also discuss the practical challenges facing the integration of such an approach into the sensor application stack.*

Keywords wireless sensor networks, context-aware systems, data-centric architectures

1. Introduction

Wireless sensor network mainly functions as an effective information capture and processing layer in both indoor and outdoor applications. As they are becoming more and more popular, the spread of ubiquitous connectivity will involve a revolution on the same scale as the Internet in the 90's. Currently the Internet consists of two layers: stable core (routers, servers) and a quickly growing fringe (personal computers, mobile devices). However, the new embedded devices will form a new layer i.e. *Internet of Things (IoT).* As they become IP-enabled, they will add a new dimension to the Internet offering exciting possibilities and challenges at the same time.

There are at least two limitations that can be easily identified in current wireless sensor networks. First of all, sensor nodes are usually devices with limited resources such as battery, memory, processing capability, and achievable data rate. An individual node is capable of sensing its environment, processing the information locally, or sending it to one or more destinations through a wireless link. This wireless link is usually short-range transmission due to a low RF transmit power. In addition, this transmission can often be disrupted by other wireless transmissions.

The second issue is related to the entire WSN application construction and installation process. Many of the current WSN deployments are application-centric: they are designed for one specific scenario with a defined *Quality of Service (QoS)* requirement. In addition, provided services are usually tightly-coupled [1]. As a result, it is very difficult to perform regular operations, such as upgrade or extension of the given installation. It is also difficult for an application developer to use data in external systems, because the protocols and services are usually closed.

We think that, in order to tackle those issues, we should model and describe the WSN system using data-centric approach. Fundamentally data drives the *Internet of Things* realm — as we stated earlier the main function of that layer is to collect and process data. So, as many researchers suggest, this seems to be the most natural way of describing WSN systems [3, 4, 16, 13]. Sensor nodes can publish information in compliance with *'data-dictionaries'* which include the metadata (data semantics or *QoS* targets) needed to use it effectively. When information is needed participants can subscribe to it with an appropriate *QoS.* As a result all information regarding state of data is published and available to use in the entire system (see chapter 2 for a more detailed explanation of the data-centric architecture).

Once the system will be constructed it will give an opportunity to create new types of distributed systems. Currently devices are getting more intelligent and powerful (in terms of processing power and memory capacity). Their usage is also getting more and more pervasive. This leads to the construction of a distributed system — heterogeneous and fully build in the physical world. Building the system is challenging, because they consist of independently developed software and hardware components. We think that a data-centric approach is fully up to the task of implementing the distributed *IoT* environment.

In addition, context information can improve and extend the overall functionality of the data-centric system. The new available data will enrich not only the application itself but also data-centric middleware. Firstly, the application will have access to more data (e.g. location or data gathered by other sensors in heterogeneous environment) which can enrich its functionality deliver do end users. Secondly, data-centric middleware can optimize some aspects of its own functionality (e.g. adaptation of information distribution) as well as usage of limited resources (e.g. bandwidth). In addition, context information can extend data-centric abstractions, such as metadata or *QoS* definition, in order to deliver a more holistic view of the physical environment to application developers.

This paper explores the case for enriching WSN environment with context data. We discuss how the context-aware sensor networks can function more effectively compared to pure data-centric sensor networks. We build our argument using examples, some of which are taken from other work, while others are developed herein. We introduce the idea of enriching the data-centric WSN environments with context information. We also discuss issues related with integration of such a design to general sensor application stack.

2. Background

More generally, a data-centric approach in application development, emphasize data usage in system design. As stated earlier, data drives all functionality of the information system. Anything else — coding, APIs, messages, packets or protocols — is potentially an unnecessary waste of focus and we should minimize its importance as much as possible. The data source can be represented as an independent *sub-system* and the real problem is to integrate those *sub-systems*.

An effective approach is to design a data-centric architecture that can link *systems of systems* into a coherent whole. We can define the following principles of the data-centric design [12]:

- Information is described by data and metadata. By the term data we understand the real world information available in the system, while the structure of data and interface is defined by the metadata. By using metadata communicating parties information in the system can be encoded and decoded.
- The behavior of the system is hidden. Participants in the data-centric system do not expose any information about their behavior or state. Only data are distributed.
- Data bus connects data producers to consumers. As the system is loosely coupled and consists of independent parties, data handling should be performed by data bus. The bus is shared among participants and is responsible for data management and distribution.
- The data-handling contracts should describe data transfers among communication parties. The contracts should be then applied by data bus and proper data

flows should be then established. The contracts should contain the *Quality of Service (QoS)* properties.

The data-centric architecture is supported by the *Object Management Group Data Distribution Service (OMG DDS)* [15] specification.

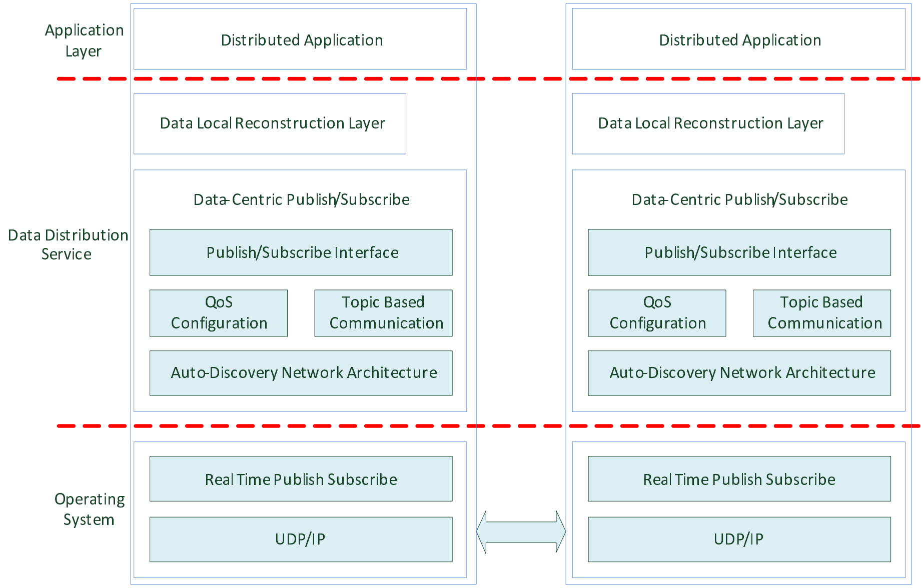
Application Layer
Data Distribution Service
Operating System
Distributed Application
Data Local Reconstruction Layer
Data-Centric Publish/Subscribe
Publish/Subscribe Interface
QoS Configuration
Topic Based Communication
Auto-Discovery Network Architecture
Real Time Publish Subscribe
UDP/IP
Distributed Application
Data Local Reconstruction Layer
Data-Centric Publish/Subscribe
Publish/Subscribe Interface
QoS Configuration
Topic Based Communication
Auto-Discovery Network Architecture
Real Time Publish Subscribe
UDP/IP

Figure 1. *DDS* is build on the top of the operating system, publish-subscribe operations are supported by the *Topics* and *QoS* definitions. It also defines additional services such as *Auto-Discovery.*

The *DDS* defines a middleware for development of data-centric publish-subscribe applications. The standard is widely used in the high-performance, mission critical networks especially in the military and air-traffic management systems. It is the official standard for publish-subscribe messaging approved by the *U.S. Department of Defense (DoD)*. The standard consists of two layers (see Figure 1):

- *Data-Centric Publish Subscribe (DCPS)* — is responsible for efficient information distribution based on the publish-subscribe paradigm.
- *Data-Local Reconstruction Layer (DLRL)* — is an optional layer that defines object interface for basic *DCSP* functionalities.

The standard itself has many successful industry implementations and deployments [6]. In addition *tinyDDS* [3] attempts to provide WSN implementation of the *OMG DDS* standard.

WSN environment can benefit from data-centric architecture in two ways. First of all, it can ease the integration process. *IoT* environments will be very heterogeneous and some of the deployed solutions will be proprietary. In order to tackle the integration issues we should focus on the data exchange between sub-systems that forms our environment. Those sub-systems can later be mapped as data-sources. Secondly, the data-centric approach gives abstraction of an individual sensors as well. Main function of sensor is to operate on information — which can be easily mapped to concepts of data and metadata.

Context in general can be described as the set of information that characterize situation of some entity [7]. The situation can be physical (e.g. location of a person) or functional (e.g. current task). In more general it can be categorized in four dimensions: *computing context*, *physical context*, *time context*, and *user context* [4]. In a pervasive environment physical context is the most important — it represents all physical parameters of executing environment of node surroundings. The most popular example includes node location but also other properties, such as temperature, speed or noise level, might be included. Because the physical context describes the physical phenomena it is very prone to errors (see chapter 3.3 for other aspects of data quality). Another important dimension is *time context.* In general data in the sensor network can be *sporadic* (e.g. events send when something happens) or *periodic* (e.g. when data are send in some more predictable way, usually regular). They can also build more complex events (e.g. control video quality based on the specific events) [4]. In the case of a sensor network there is also one important aspect of *time context.* When data is available it needs to be available in a timely manner (*near-real-time*). Not everything has to be accessible in *real-time*, but we should have reliable information about predicted time of data delivery. In order to achieve that we should add additional constrains to the service level. By enriching the environment with context information we can add additional dimension to WSN. Context data can influence data-centric WSN systems by extending the functionality of metadata as well as *QoS* properties. Because all information will be present in the shared data bus — it can be utilized in the entire WSN environment.

3. Opportunities

The following section and simple examples revolve around the same main theme, potential benefits of enriching wireless sensor networks with context information.

3.1. Throughput optimization

In recent years IEEE 802.11-based networks have enjoyed wide adoption — they are mostly deployed in order to provide Internet access. In indoor environments, the *WiFi* network, sometimes coexists with IEEE 802.15.4-based environments. In addition to location neighborhood they also share spectrum — both operate on unlicensed 2.4 GHz ISM band. When operating in the same or adjacent frequencies, 802.11 and 802.15.4 radios can interfere with each other. In particular, only 4 among all 16 channels of 802.15.4 do not overlap with the channels of 802.11 (see Figure 2). As many studies show the coexistence have positive and negative effects.

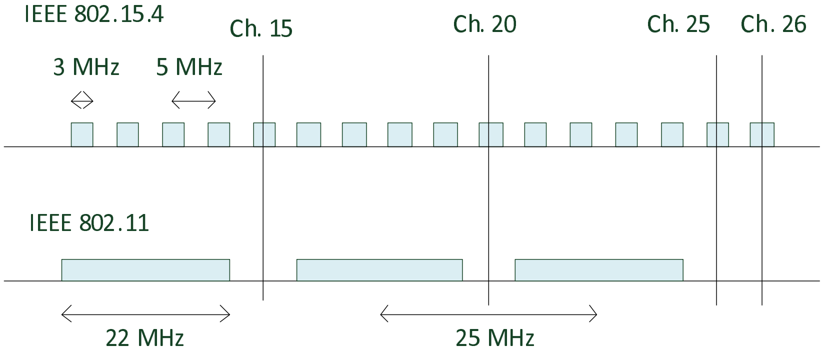

Figure 2. Both 802.11 and 802.15.4 radio standards share the 2.4 GHz ISM band. Only a few of the 802.15.4 channels do not interfere with *WiFi* channels.

On the positive side we can mention improvements in energy effectiveness. In order to achieve seamless mobility experience *WiFi-enabled* devices must actively scan for new *WiFi access points*, once they leave the coverage of the current network. However, this is energy consuming. One of the solution is to use 802.15.4 radios to identify the unique interference signatures generated by *WiFi* signals. It will allow to detect the existence of *WiFi APs* in a passive manner [17].

Network coexistence also has negative effects. One of them is impact on throughput. *ZigBee* performance under *WiFi* interference has been measured in both empirical and theoretical experiments. According to studies performed by *ZigBee Alliance*, *ZigBee* networks can coexist with *WiFi* while still maintaining the desirable *QoS*. In terms of throughput *ZigBee* does not affect IEEE 802.11g significantly (the influence on 802.11n is not discussed widely). However, the throughput of *ZigBee* drops when the chosen channels of operation overlap.

As more and more devices are put into work, the overlap will happen more often. One way to tackle it is to use static configuration of channels in the working environment. However, in addition to normal work patterns, there are two other things that need to be considered:

- Devices are not used all the time in the same intensive way. There are often peaks in the radio bandwidth usage. Lets analyze a scenario in which temperature monitoring sensor network is deployed in an office. During working hours traffic is huge, and the bandwidth is heavily used. On the other hand, outside working hours — when there is no traffic generated by users, the situation is different. The communication is not disrupted by other parties,
- In addition the standard packet base transitions (such as 802.15.4 or 802.11) there are also other devices that uses the 2.4 GhZ ISM bandwidth. We can mention here microwave ovens or electronic baby monitoring devices. All of them can influence the 802.15.4 transitions — and their communication is temporary (they are not

working all the time). There are some periods during a day when communication is heavily disturbed by this kind of influence.

As a result static channel configuration is not good solution [9].

One of the solutions to improve the throughput is to dynamically adapt to changes in radio frequency quality. However, we need to first acquire the bandwidth occupancy data. There are two ways we can tackle this issue — both use the physical context of the communication sensor node. First approach is to leverage existing 802.15.4 collision avoidance mechanisms and the second approach requires an external system for handling the radio quality data acquisition.

Regarding the first approach, currently there are 2 mechanisms to avoid all the nodes start emitting at the same time in the 802.15.4: *Carrier Sense Multiple Access-Collision Avoidance* (*CSMA-CA*) and *Guarantee Time Slots* (*GTS*). While the latter requires the existence of a centralized node which manages the process — it is not good for scalability issues, we will focus on the first one. In the first method each node listens to the medium prior to transition. If the energy found higher of a defined level (on a single channel) the node the transceiver waits during a random time (including in an interval) and tries again. When we are performing an energy scan we can focus on both energy level (overall radio activity) or 802.15.4 transitions on a single channel. However, the described mechanism is applied in a single (or several channels) prior to using it. There are no mechanisms for dynamic channel adaptability for the entire network in the 802.15.4 standard.

The second approach focuses on external system usage for gathering context data. The mentioned systems are capable of scanning radio frequency in search for interference and location of radio signal sources. They are usually use as an assistance in the *WiFi* deployments and help solve configuration and performance issues [5]. In addition to their normal function, they can also be used in WSN deployments. Gathered information about the working environments (particularly radio frequency usage) can enrich context and help choose and define optimal radio channels for radio communication. In addition, the mentioned context-aware systems will have view of entire WNS deployment (not only nearest neighborhood).

3.2. Location estimation

One of the most important aspects of the WSN application is to identify the location of its sensor. When an alarm is raised we want to know what is the location of the sensor in order to identify the source of the alarm. After that we can respond to the event properly. Another important issue related with localization is mobility of sensor nodes. Most of the current WNS deployments are static, however, in future due to increased popularity and the ubiquitous nature of sensor networks it is important to assume that sensors can easily move. Mobility support should be seamless to devices and they should be aware of it. Another aspect of mobility lies in the access control of its applications. For example, *Patient Monitoring Networks (PAN)* are more becoming more and more popular. Their main function is to sense the vital signs of patients. They becoming part of a bigger *Medical Sensor Networks (MSNs)*. This type of sensor networks can be deployed in many diverse organizations such as fitness centers, hospitals or retirement homes. They allow pervasive health monitoring on many environments and also locations. However, this raises an issue related with the access to the data, e.g. who should have access to the data gathered by *PAN* [8].

The location problem has been in the literature for quite a long time and many solutions and algorithms have been proposed [10]. Currently, the most popular is the *two-step positioning* approach (see Figure 3). The localization process can be divided into two phases. In the first phase, some signals are transmitted between the target node and a number of reference nodes. During this process, some properties of these signals, such as arrival time, signal strength, and direction are captured by the receivers. In the second phase, based on signal parameters we determine physical position. In the latter phase we can optimize and improve the accuracy of the results.

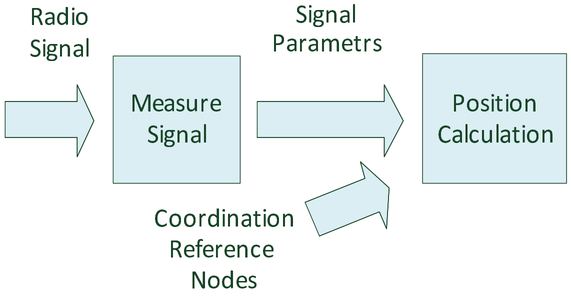

Figure 3. Two-step positioning approach: a) we extract certain signal parameters (e.g. *Time of Arrival* or *Received Signal Strength*), b) we calculate the position of sensor nodes.

Realization of mentioned *two-phase schema* in the WNS environment purely on nodes is costly. It requires both computer power and network bandwidth. In addition, there are always issues related with the accuracy of measurements. As a result, location process is usually performed by the central unit in the network and only signal parameters are gathered by WNS [10]. But still, measurements and initial processing of data should be done locally.

However, because location is physical property, the information about position is already present in the context of physical environment. So, we can extend the WSN environment with the required data. We can obtain this data through location-aware systems. The main task of those systems is to develop a *world model* of artifacts from a physical environment (e.g. devices or people) and later derive context from the constructed model. Model can be build using different methods, we can track artifacts based on a camera pattern recognition algorithm, user-assisted mechanisms or, as mentioned earlier, techniques based on radio signal monitoring. All the data gathered by the system can enrich WSN environment. As a result, an individual sensor will obtain the more accurate location data from external systems (more sophisticated algorithms can be applied in order to minimize errors) allowing it to be more independent. The handling of mobility of nodes can be improved as well because information about changes in node position can be accessible through the sensor context.

In addition to context extension, WSN can extend the location-aware system. They can contribute to a model update process. As researchers suggest the process of maintain the world model is important, because it needs to be accurate with the physical world [11]. WSN can contribute by sending other context information that can indicate change in the physical environment. For example, if an accelerometer of a sensor is detecting changes in its state it can indicate that its location might have changed. So the model can be updated accordingly. In addition to updating the direct state of WNS, a sensor can also help describe the physical environment more accurately. For example, if the temperature reading of a sensor drops dramatically it may indicate that the window was opened. As the result world model should be updated in order to properly and accurately describe the physical environment.

3.3. Middleware architecture

Another important aspect in the *IoT* vision is the construction of scalable robust middleware. Middleware is usually responsible for handling the heterogeneity of environment and it provides abstraction for software developers. In addition, we think that the middleware should also be responsible for handling context distribution.

Heterogeneity is caused by the continuous development and deployment of new devices and new services. As more sensors are available, the level of computational power, a network density and lifetime increase. For example, a network consisting of powerful devices can perform complex data analysis but cannot be deployed in big numbers over large area or run for a long period of time due to cost and energy constrains. On the other hand, network consisting of just weak devices cannot perform complex operations. By integrating a different type of sensor heterogeneous WSNs can perform more sophisticated operations over a larger region and longer interval than is possible with homogeneous WSNs.

A defined set of abstraction is also important if we think about the wide adaptation of *IoT*. Traditionally, developers would prepare an application for a specific hardware platform with special focus on energy and memory efficiency. As a result, there are currently many programming models created especially for the WSN [14]. Unfortunately this programming approach is no longer feasible, due to the increased heterogeneity in the network. Instead, an application should use platform independent abstractions. As we mentioned in chapter 1, we think that the WSN environment should be represented using the data-centric approach. The architecture defines a set of abstractions that describe WSN in its natural way of operation. Because the *OMG DDS* standard defines data-centric architecture across many platforms we can think about employing it as the solution to deal with network heterogeneity.

Context-awareness becomes a very important issue in the *IoT* deployment. It can be broadly defined as the ability to provide services with full the awareness of the current execution environment. The process of production, processing and delivering of the context information is known as *context data distribution.* In general, context information has to be delivered to all interested parties in order to be fully usable. Because of the mentioned requirement the middleware has to transparently and timely transfer a huge amount of data. Also, additional factors, such as system scalability and reliability, has to be taken into account. When both these conditions are met the single node can fully use context information.

Additional challenges appear when we want to include external sources of the context information. The systems can produce a huge amount of data that needs to be filtered or processed. This also increases the need for load balancing inside the WNS system. One if the key benefits of the presence of such external sources of information is the level of high quality data it can gather and deliver. When we think about *context data distribution* some additional concepts regarding *quality of data* should be taken into account. The main reason for this is, that data with poor quality can completely mislead the application. In addition, transfer of low quality data can reduce available bandwidth.

Concept of *Quality of Context (QoC)* is rather new in the research community. It was introduced in order to ensure that *"the availability of the context data with the right quality, in the right place, and at the right time"* is taken into account [2]. *QoC* recognizes and consists of both: data quality and process of distribution of the data. Data quality in WNS has many dimensions: it has to be valid (data should be send in proper form or standard), up-to date (the data has to be correct in given time) and read with proper precision (data gathered by sensors may have a different precision, e.g location information obtained through *GPS* is not so accurate as *WiFi-based*). The process of data distribution is very important in the mobile and ubiquitous WSN environment, because sensors are usually distributed over a geographical area and connectivity is realized by the low-powered radios. As the result process of data distribution should be asynchronous and should support mobility. In addition, in order to manage different *QoS* across an entire system, *QoS* management services have to be introduced [13]. These services will be responsible for managing tradeoffs involved in providing aggregated *QoC* in the system.

Another important issue is representation of context. Representation can be domain specific or can be described by general models [2]. In the second category we can include *key-value*, *markup scheme*, *object-oriented* or *ontology-based* models. We think that object representation is best suited for the WSN environment. As we stated in chapter 1, the main function of the sensor system is exchange of data. The mentioned information usually consist of a raw data gathered by sensor nodes. However, the data can be later extended by the context information. We think that extensibility support available in the object-oriented models the best fit to our requirements. In addition, the object model is supported by the *OMG DDS* standard — metadata are defined by objects. In order to support *QoC*, we can extend the metadata by the *Quality of Service (QoS)* definitions. *OMG DDS* provide a huge set of already defined *QoS* parameters, e.g. reliability (ensures that data is delivered) or durability and lifespan (controls persistence, how and where data is stored) [15]. We think that by building data-centric middleware based on the *OMG DDS* we can create middleware capable of performing context data distribution for future *IoT* environment.

4. Future research directions

Context-aware services can provide data which can enriching WNS networks. However, the process raises many important issues in the current sensor application stack. Once the information will be broadly available a different part of the stack can leverage their presence. A different part of the current application sensor stack will be influenced (see Figure 4). Data link layer can benefit from more accurate measurements of air quality, so the overall throughput and packet loss can be improved. As a result, more reliable communication can be provided. In addition to the data link layer, the network layer can be optimized as well. Currently, wireless link has been designed using wired link the blueprint — wireless channels are realization of point-to-point communication, and routing graphs are based on this abstractions. However, wireless medium is different — it supports mobility and portability. Once we will know the geographical location of communicating nodes we can organize routing paths accordingly.

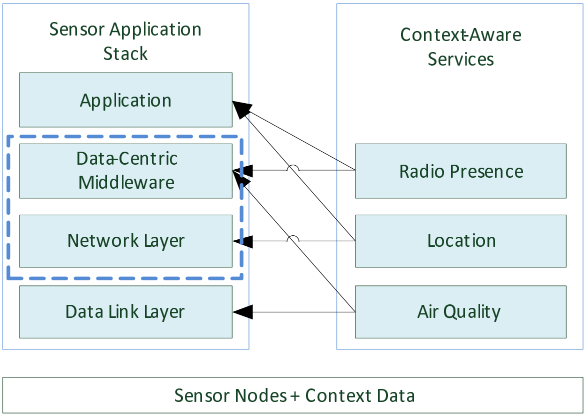

Figure 4. Once the context data will be available, different parts of the current sensor application stack will be affected (e.g. data link layer can leverage more precise radio quality information) as well as different services will be enabled (e.g. monitoring of data gathered by the WSN).

The middleware layer can also benefit from context data. Firstly, *QoS* can be enriched by the information about air and link quality. Secondly, information about location will be available to all communicating parties. Thirdly, information about the presence of the new radio sources can improve the discovery services in the network. As a result, the entire *Data Distribution Service* can benefit from the presence of context data. In addition, based on context information, the application layer can benefit as well. For example, more robust WSN monitoring can be built. Unfortunately, realization of mentioned opportunities is challenging because it requires changes in many areas of the sensor stack.

Another important challenge lies within the adaptability of the context dissemination process. In general the node is responsible for delivering data to all interested parties. Currently, we can distinguish the following different categories of dissemination processes: *flooding-based*, *direct access*, *selection-based* and *gossip-based*. All of them have positive and negative properties (e.g. accuracy of context data or bandwidth and memory usage). Adopting the dissemination process in the existing deployment is very challenging. For example, a sensor network with a mobile sensor could use *gossip-based* protocols to enable context propagation in geographically distant areas, and could use *selection-based* protocols during close proximity distribution, when only required data are send [2]. The process of adapting the dissemination process is important because it will improve system effectiveness. Because information about the location and link quality as well as location will be available to nodes, dissemination protocol can be chosen by middleware based on the accurate physical data. As a result, the adaptation process can be improved by the context data.

Directly connected with data dissemination is the process of data gathering or aggregation [16]. Sometimes data should be grouped if certain *QoS* requirements are met. For example, if we are interested in room-based location we can perform an aggregation of nodes positions in order to describe the context of occurred events. The process of selecting criteria for better aggregation is important. Currently, there are many mechanisms available for data gathering. These mechanisms take into account many factors (e.g. energy efficiency or proximity of the node [16]). Similar as in case of the data dissemination, the process of adapting the aggregation process is important because it will improve system effectiveness. We think that context information can help adapt the data aggregation process to current physical conditions.

5. Conclusion

Context data can enrich the WSN environment in many ways. In this paper we presented two types of context information which have the greatest potential: location and bandwidth quality. In addition, we mentioned specific use cases where context data can be used. However, presented opportunities come with many challenges (e.g. middleware architecture). In our future research we will try to address some of the issues presented in our paper. We believe that the future *IoT* environments will fully benefit from the presence of context information.

References

[1] Akyildiz I. F., Melodia T., Chowdhury K. R.: A survey on wireless multimedia sensor networks. *Computer Networks*, 51(4):921 – 960, 2007.

[2] Bellavista P., Corradi A., Fanelli M., Foschini L.: A survey of context data distribution for mobile ubiquitous systems. *ACM Comput. Surv.*, 45(1), April 2013.

[3] Boonma P., Suzuki J.: Middleware support for pluggable non-functional properties in wireless sensor networks. In *Proc. of the 2008 IEEE Congress on Services — Part I*, SERVICES '08, pp. 360–367, Washington, DC, USA, 2008. IEEE Computer Society.

[4] Chen G., Kotz D.: *A survey of context-aware mobile computing research.* Technical Report, Dept. of Computer Science, Dartmouth College, 2000.

[5] Cisco.: *Context Aware Mobility (MSE) API.* Cisco Live Conference, 2011.

[6] Corsaro A.: *Efficient and Extensible Data-Centric Architectures.* PrismTech OpenSplice DDS Webcasts, 2011.

[7] Dey A. K., Abowd G. D., Salber D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications, 2001.

[8] Garcia-Morchon O., Wehrle K.: Modular context-aware access control for medical sensor networks. In *Proc. of the 15th ACM symposium on Access control models and technologies*, SACMAT '10, pp. 129–138, New York, NY, USA, 2010. ACM.

[9] Geirhofer S., Tong L., Sadler B. M.: Dynamic spectrum access in wlan channels: empirical model and its stochastic analysis. In *Proc. of the first international workshop on Technology and policy for accessing spectrum*, TAPAS '06, New York, NY, USA, 2006. ACM.

[10] Gezici S.: A survey on wireless position estimation. *Wireless Personal Communications*, 44:263–282, 2008. 10.1007/s11277-007-9375-z.

[11] Harle R., Hopper A.: Towards autonomous updating of world models in location--aware spaces. *Personal and Ubiquitous Computing*, 12:317–330, 2008. 10.1007 /s00779-006-0103-6.

[12] Joshi R.: Data-centric architecture: A model for the era of big data. *Dr. Dobbs*, 3.2011.

[13] Loyall J. P., Carvalho M., III A. M., Schmidt D., Sinclair A., Gillen M., Edmondson J., Bunch L., Corman D.: Qos enabled dissemination of managed information objects in a publish-subscribe-query information broker. *Defense Transformation and Net-Centric Systems 2009*, 7350:73500M, 2009.

[14] Mottola L., Picco G. P.: Programming wireless sensor networks: Fundamental concepts and state of the art. *ACM Comput. Surv.*, 43(3):19:1–19:51, April 2011.

[15] Standard.: *Data-Distribution Service for Real-Time Systems (DDS) v1.2.* Object Management Group (OMG), 2007.

[16] Stojmenovi I., Olariu S.: *Data-Centric Protocols for Wireless Sensor Networks*, pp. 417–456. John Wiley and Sons, Inc., 2005.

[17] Zhou R., Xiong Y., Xing G., Sun L., Ma J.: Zifi: wireless lan discovery via zigbee interference signatures. In *Proc. of the sixteenth annual international conference on Mobile computing and networking*, MobiCom '10, pp. 49–60, New York, NY, USA, 2010. ACM.

Affiliations

Marek Konieczny
AGH University of Science and Technology, Faculty of Electrical Engineering, Automatics, Computer Science and Electronics, Department of Computer Science, al. A. Mickiewicza 30, 30-059 Krakow, Poland, marekko@agh.edu.pl

Received: 23.03.2012
Revised: 9.07.2012
Accepted: 3.09.2012