

Research Report

e-SENSE Protocol Stack Architecture for Wireless Sensor Networks

W. Schott and U. Hunkeler

IBM Research GmbH
Zurich Research Laboratory
8803 Rüschlikon
Switzerland
e-mail: (sct, [hun](mailto:hun@zurich.ibm.com))@zurich.ibm.com

A. Gluhak, M. Presser, and R. Tafazolli
Center for Communication Systems Research
University of Surrey
Guildford, GU2 7XH
UK
e-mail: (a.gluhak, m.presser, [r.tafazolli](mailto:r.tafazolli@surrey.ac.uk))@surrey.ac.uk

LIMITED DISTRIBUTION NOTICE

This report has been submitted for publication outside of IBM and will probably be copyrighted if accepted for publication. It has been issued as a Research Report for early dissemination of its contents. In view of the transfer of copyright to the outside publisher, its distribution outside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies (e.g., payment of royalties). Some reports are available at <http://domino.watson.ibm.com/library/Cyberdig.nsf/home>.



Research
Almaden • Austin • Beijing • Delhi • Haifa • T.J. Watson • Tokyo • Zurich

e-SENSE Protocol Stack Architecture for Wireless Sensor Networks

W. Schott, A. Gluhak, M. Presser, U. Hunkeler and R. Tafazolli

Abstract—The e-SENSE project has the vision to enable new context-aware applications and services in beyond 3G mobile communication systems by providing the technology required to capture the context surrounding the service users and service-related objects through wireless sensor networks. This paper introduces a flexible protocol stack architecture for wireless sensor networks, which allows the incorporation of novel connectivity, management, and middleware concepts into the e-SENSE system, and thus provides advanced support functionality and high adaptability required for a wide range of application and deployment scenarios.

Index Terms—Wireless Sensor Networks (WSNs), Protocol Stack, Sensor Connectivity, Middleware, Context Awareness

I. INTRODUCTION

SENSOR network research has gained significant attention in the research community in recent years. As a consequence, a variety of protocols for wireless sensor networks (WSNs) have been developed by researchers across the world, addressing different issues such as the design of advanced medium-access control, networking, and data transport protocols, as well as the development of efficient data messaging concepts. More recently, research work has also been started to address the compatibility and reconfigurability issues of the resulting overall protocol stack. New protocol architectures for WSN have already been proposed to unify the stack executed on the processor in sensor nodes [2][3], or to more easily adapt the stack to application requirements [4]. While the initial proposals seem promising, the work is still in the early stages.

The current de-facto standard for operating low-cost, low-power devices in a WSN has been specified by the ZigBee Alliance [5]. Based on the IEEE 802.15.4 MAC and physical layer standard [6], the ZigBee specification defines an architecture for sensor networks that comprises a network layer, an application support layer, as well as a

security managing unit. This specification represents a first important step towards an architecture that ensures compatible implementations of wireless sensor nodes and networks, and allows a unified development of sensor applications. While being designed for mainly simple, static sensor and control applications, the current version of the ZigBee protocol stack is constrained by limited protocol options and a lack of service and support functions. Therefore, a more flexible architecture is required to satisfy the requirements of new applications, which will likely emerge with the integration of WSN into beyond 3G mobile communication systems.

In this paper, we present a novel flexible protocol stack architecture for WSN. It has been developed as part of the IST project e-SENSE [1]. This protocol architecture creates an architectural framework that

- is able to easily integrate advanced protocol concepts and new communication paradigms, e.g. data centricity,
- is flexible enough to allow the system to be configured for various application requirements by selecting appropriate protocols and service functions,
- provides a system infrastructure to allow efficient operation of the protocols through cross-layer interaction,
- allows wireless sensor nodes and networks to adapt their behaviour according to environmental conditions, and
- offers enhanced support functionality required for many applications, e.g. localisation.

II. OVERVIEW OF THE E-SENSE PROTOCOL STACK

Figure 1 shows the architecture of the e-SENSE protocol stack. The architecture is divided into four logical subsystems, namely the connectivity (CO), middleware (MI), management (MA), and application (AP) subsystem. Each subsystem comprises various protocol entities, which offer a wide range of services at various service access points (SAPs) to other subsystems. The entities can be combined in many ways to configure the protocol stack according to the role of the sensor node and application requirements. The entire stack has to be implemented only in a full-function sensor device, while fewer functions are required in a reduced-function device. Note that the protocol stack is not as strictly layered as specified in the classic OSI model to facilitate cross-layer optimisation.

The *connectivity subsystem* of the e-SENSE protocol stack consists of functions that are required for operating the physical layer (PHY), the medium access control (MAC), and the network (NW) layer. The PHY functions comprise the

Manuscript received January 30, 2007.

W. Schott and U. Hunkeler are with the IBM Research GmbH, Zurich, Switzerland (e-mail: {sct, hun}@zurich.ibm.com)

A. Gluhak M. Presser and R. Tafazolli are with the Center for Communication Systems Research, University of Surrey, Guildford, GU2 7XH, UK (e-mail: {a.gluhak, m.presser, r.tafazolli}@surrey.ac.uk).

This paper describes work undertaken in the context of the e-SENSE project, "Capturing Ambient Intelligence for Mobile Communications through Wireless Sensor Networks" (www.ist-e-SENSE.org). e-SENSE is an Integrated Project (IP) supported by the European 6th Framework Programme, contract number: 027227. The authors would like to acknowledge the contributions of their colleagues from the e-SENSE Consortium.

radio transceiver. The MAC functions control the access of nodes to the shared radio channel and provide means for synchronisation, beaconing, and reliable message transmission and reception. It is initially assumed that the e-SENSE PHY and MAC functions are very similar to those defined in the IEEE 802.15.4 low-rate wireless PAN standard [6]. The NW functions can create and maintain WSNs. WSNs consist of various types of network nodes, namely coordinators, clusterheads, and end devices, and can support a star, tree, or mesh network topology. The NW functions also include mechanisms to discover existing networks, to let devices join and leave a network, to discover and maintain routes between nodes, and to transfer messages securely and reliably from a source to the destination.

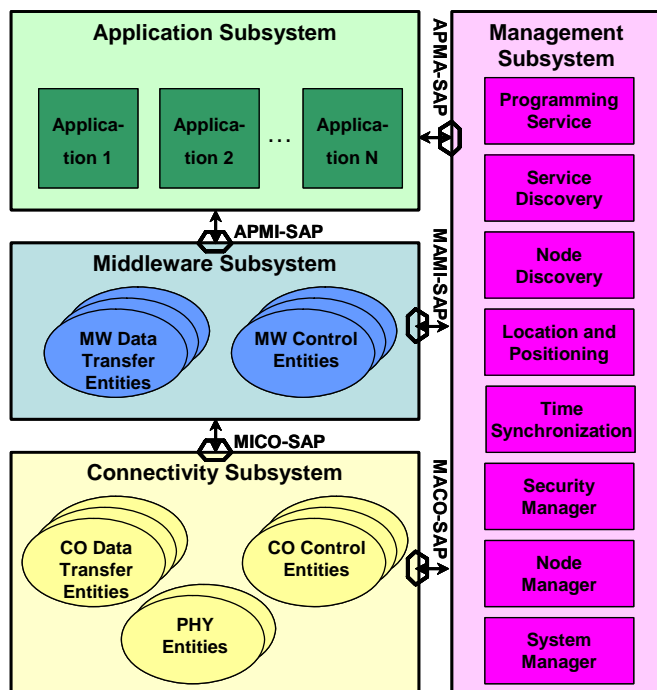


Figure 1: Architecture of the e-SENSE protocol stack

The *middleware subsystem* has the purpose to create and maintain an infrastructure network where information sensed by nodes is processed in a distributed fashion and, if necessary, the result is transmitted to an actuating node or to a backbone network by means of a gateway. The middleware subsystem provides two types of data transfer services for the transport of application data packets, namely a node-centric service and a data-centric service. The former service enables sensor nodes to transfer data to particular nodes or group of nodes by using an explicit address. The latter one is a data transfer service based on a publish/subscribe paradigm. This messaging system facilitates information exchange among nodes because senders and receivers do not communicate directly with each other but via a message broker. In addition to the transport mechanisms for sensor information, the middleware subsystem also provides means to process the sensed information in a distributed fashion.

The *management subsystem* is responsible for the

configuration and initialisation of the connectivity and middleware subsystems based on information provided in the application profiles. The management subsystem interfaces to the middleware subsystem to configure the middleware and to exchange messages with peer nodes, and interacts with the connectivity subsystem to manage the NW, MAC, and PHY functions of the sensor node. In the management subsystem, several functions are implemented: service and node discovery, location and positioning, time synchronisation, security manager, node and system manager. The service and node discovery functions enable the subsystems to find a service or node in the network. The location and positioning functions supply information on the absolute or relative position of the node in space. The security manager controls all issues related to authentication, privacy and trust, while the node and system manager handles all tasks related to the correct operation of the node and system. The management subsystem also provides means to program and manage the life-cycle of the e-SENSE system by updating software code according to application requirements; moreover, it comprises an information database that stores all parameters and attributes related to the subsystems.

The *application subsystem* hosts one or several sensor applications. Each application can send and receive sensor data by using the data transfer services of the middleware subsystem. The application subsystem configures the management subsystem of the node according to its functionality and role in the sensor network.

III. CONNECTIVITY SUBSYSTEM

This section presents further details on the services and corresponding protocol functions of the connectivity subsystem of the e-SENSE protocol stack.

A. Data Transfer Service

The connectivity subsystem provides a data transfer service to the middleware subsystem at the MICO-SAP. This service can be used to exchange service data units (SDUs) between two or more sensor nodes within the WSN.

After invoking this service by requesting the transmission of an SDU to a destination node, a suitable protocol profile is selected according to the specified QoS requirements. This profile defines the routing and transmission method to be used for the data transfer. Accordingly, a sequence of proper operations such as encryption, segmentation, and coding is performed on the SDU to create packets that are well-suited for hop-by-hop transmission to the destination. The address of the next-hop node is determined by using either a pre-generated routing table or by applying opportunistic routing principles. After performing these operations, MAC and PHY functions are invoked to control the access of the node to the radio channel and to transmit the packet to the next-hop node. Retransmission of packets may be applied in case of transmission failures.

The connectivity subsystem of the next-hop or destination node can only receive data packets after a proper reception

profile has been selected. According to this profile, the connectivity subsystem can activate its transceiver to receive packets addressed for this node or can actively poll data from a specified source node. Moreover, reception filters can be installed to receive packets only from particular source nodes or evaluation handlers can be registered for data-centric processing of received data packets. After the receiver chain has been configured, the packet can be received, decoded, and further processed based on the protocol discriminator information provided in the received packet and the reception profile. If the receiving node is the final destination, the reconstructed SDU is indicated to the middleware subsystem; otherwise, the received packet is forwarded to the next-hop node.

B. Configuration and Support Services

The connectivity subsystem provides configuration and support services to the management subsystem at the MACOSAP. These services provide means to establish and control a WSN, to configure the protocol stack, and to set and monitor protocol parameters as well as internal states of the connectivity subsystem according to given application requirements. Next, the main services are briefly described.

Network Formation: This service configures the connectivity subsystem of a full-function device so that a new network can be established. To do so, the connectivity subsystem performs an energy detection scan followed by an active scan over all radio channels within the personal operating space (POS) of the node to identify the best-suited radio channel for the network. After selecting a channel, a unique network identifier (WSNid) is assigned and functions are started to control the access of network nodes to the radio channel. These functions can be provided with or without periodically broadcasting beacon frames.

Network and Device Discovery: These services discover networks and devices, which are currently operated within the POS of the node. To discover an established network, an active scan is performed over all specified radio channels to retrieve information provided in beacon frames transmitted by the coordinators or clusterheads of existing networks. This information contains addressing information, the WSNid, security settings, and information on whether the beaconing device permits joining. All parameters are stored into the neighbour table of the information database. The device discovery service can be used to iteratively search for neighbour nodes. The discovered device identifier as well as device attributes such as distance, network role, etc., are reported to the service user.

Network Join and Leave: These services provide means to join or remove an end device to or from an existing network. The join service establishes a parent-child relationship between the node that requests joining the network and the coordinator or clusterhead of a selected WSN. The child node initiates the association process by selecting a suitable parent node from its neighbour table and issuing to this node an association request command indicating its own operational

capabilities. After receiving this command, the parent node may allow the child node to join the network if sufficient radio resources are available. If this is the case, a new entry is created in the neighbour table of the parent node, containing information on the capabilities of the joining device. Moreover, a unique network-address is assigned to the new node and sent back to the child with an association response command. The child node acknowledges the successful reception of the command and terminates the association process after having stored the assigned network-addresses into its information database. A similar service is provided to remove nodes from the network; however, the required disassociation functions can be initiated by the end device, the network coordinator, as well as the associated clusterhead.

Start Clusterhead: This service configures the connectivity subsystem of a full-function device, which have already joined the WSN, to start its operation as a clusterhead. The service verifies whether the device has already joined the network, configures the MAC frame structure according to the needs of the network manager, broadcasts beacon frames, and controls the association and disassociation process of a joining or leaving end device. Moreover, this service can also be used to reconfigure the MAC frame structure of the network coordinator and already operational clusterheads.

Synchronisation: This service of the connectivity subsystem establishes and maintains timing synchronisation between an end device and the coordinator or clusterhead in a beacon-enabled WSN. After enabling the radio receiver, the synchronisation procedure starts searching for the next beacon frame transmitted on the radio channels specified for the device. Depending on the requested service, the receiver can also continuously track beacons by regularly activating the radio. Moreover, this service is also used to notify its user in case of a synchronisation loss.

Location Positioning: This service of the connectivity subsystem provides means to obtain absolute or relative location information of a device. The location information can be its geographical position, but also other metrics that are related to the location of the device such as the “signal strength”, “direction of arrival” or “time of arrival” of a received radio signal. To obtain the requested location information from the connectivity subsystem, the location positioning service verifies whether a recently updated location or metric value can be found in its own information database. If the metric value is not available or outdated, the service initiates the exchange of test frames over the appropriate radio channel to estimate the requested metric. This metric together with a parameter characterising its quality is finally reported to the service user.

IV. MIDDLEWARE SUBSYSTEM

This section presents further details on the services provided by the middleware subsystem. In distributed systems, the middleware bridges the gap between the operating system and the application to facilitate the

development of distributed applications. The services can be grouped into a node-centric data service and a context access service. The middleware subsystem further provides support for a distributed data processing service.

A. Node-Centric Data Service

The node-centric data transfer service functions are responsible for the transfer of data to a specific node or a group of nodes. They are based on the data transfer functions provided by the connectivity subsystem.

The middleware subsystem provides mechanisms for a reliable data transport that supports “guaranteed delivery”, “best effort”, and “probabilistic guarantees”. Furthermore, it can be specified whether a packet should be delivered “at least once”, “at most once”, or “exactly once”. The sender can request to be informed about the successful data delivery.

With support of the management subsystem, the middleware subsystem also provides congestion control that takes into account the priority level of the data to be transmitted. Data with the highest priority is transferred first. If bandwidth is still available, then data with the next lower priority is transferred. If data that should be delivered with a low priority cannot be delivered due to congestion, the data source is informed. The data source can then decide to reduce the bandwidth it requires by reducing the quality of the results or by lowering the sampling rate. In doing so the data gets transmitted with a higher priority. This allows an application to continue operation at a reduced quality instead of being shutdown completely due to congestion.

B. Context Access Service

The context access service maps data from a source to an interested receiver without the source or the receiver being aware of each other. It would be much too complicated for a sensor to discover and maintain a list of receivers that were potentially interested in its measurements. Similarly an application, which processes sensor data, is often not interested from where the data originates but rather that it receives a specific kind of data. Therefore, the middleware offers a publish/subscribe system which decouples the data sources (e.g. sensors) from the data sinks (e.g. applications or actuators) by using message brokers. A semantic mapping service is used to translate high-level descriptions of data and requests to low-level handlers used in the publish/subscribe system. Furthermore, the middleware offers means for processing and combining data in order to provide contextual information. Similar to the node-centric data service, the context access service supports data transport with various delivery options and priority classes to deal with congestion.

C. Distributed Data Processing

The raw data produced by the sensors are often not usable by an application. To obtain data that is more meaningful to the application, the raw data has to be processed and data from different sensors has to be combined. Important bandwidth savings could be obtained by doing such processing as close to the source as possible. In addition, an application might not

know how to process the data from the particular set of sensors available in the local network in order to obtain the data the application is interested in.

To solve this problem, the middleware subsystem of the e-SENSE system supports distributed processing in the form of task networks [7]. Each processing step, or task, subscribes to a number of topics and publishes its results on a different topic. An algorithm that implements a processing step can be made available to the network by registering it with the middleware, which in turn informs the broker about its existence. When an application subscribes to a topic for which there is (currently) no data source, the broker tries to combine processing steps in order to obtain data for the desired topic by combining data from available data sources.

V. MANAGEMENT SUBSYSTEM

This section presents further details on the services provided by the management subsystem. These services can be logically grouped into support services, management services, and programming services.

A. Support Services

The support services provide complementary functionalities to the application and other subsystems, which go beyond middleware and connectivity aspects. In the following, these services are briefly described.

Service Discovery: Sensor nodes, which want to offer their services for a given time to other nodes, have to register in a service directory. This service directory stores information about all available services in the WSN, and may dynamically move from one node to another one in the WSN. A service directory may refuse to register a service with a given service description, and may set another service expiration time than specified in the registration request; however, the service registrar receives indications about any change of its service registration until it deregisters the service. The service discovery function allows a sensor node to search for a specific service offered by any other node in the WSN by propagating a service query to the service directory. The service directory responds to the query with a list of services, including information on how they can be accessed and when they expire.

Node Discovery: This service provides network topology information on nodes of the WSN that are in the neighbourhood of a sensor node. The description of the surrounding network topology is provided as a tree-like list that comprises for each node the node identifier and its corresponding attributes.

Location and Positioning: This service enables a sensor node to obtain absolute or relative geographical location position information of all sensor nodes in the WSN. The own position of the sensor node is computed by triggering the execution of an appropriate location positioning algorithm provided in the connectivity subsystem. On request, the retrieved position information together with a parameter characterising its accuracy can be disseminated in the

network. To obtain location information of other nodes in the WSN, protocol messages are exchanged with neighbour devices to retrieve the desired position information in their information data bases. Note that the devices in the network may compute their positions using a variety of algorithms, selected on user preferences and available resources.

Time Synchronisation: This service manages all synchronisation related issues within a node and between sensor nodes. There are many synchronisation sources in a WSN which can provide timing and/or synchronisation information. Each potential source has to register its capabilities in a service directory before it can be invoked when required. The role of the synchronisation service is it to configure and control the various synchronisation sources, and provide a unified interface for sensor nodes to request and receive synchronisation information.

B. Management Services

The management services are provided by the node, system and security manager, which implement mechanisms for initialising, configuring, and securely operating sensor nodes in a WSN.

The *node manager* handles the initialisation, configuration and reconfiguration of all subsystems of a sensor node. In addition, the node manager takes care of enforcing local resource and power management policies. It also provides an information service that allows protocol entities in a subsystem to access and modify parameters of other protocol elements, possibly located in other subsystems. In this way, information can efficiently be exchanged across layers as required for cross-layer optimisation of the protocols.

The *system manager* coordinates the configuration across sensor nodes of a whole network. It assigns roles to sensor nodes and adapts/reconfigures the protocol stack executed on the sensor node according to application requirements and current network conditions.

The *security manager* controls all aspects of security across all subsystems of the protocol stack. The security manager is responsible for defining the level of security and trust. Its management decisions depend on the context information provided by the node discovery, service discovery, and localisation service, and on the information provided by the application subsystem.

C. Programming Services

Life-cycle management is an important operational aspect of WSNs as faults may be detected in original software components, the application environment may change, or more efficient algorithms may become available for specific tasks. Software components can include protocol elements, computational algorithms, or data structures of the information data base of sensor nodes. The programming service functions provide means to program and manage the software components of a sensor node by injecting code and configuration data according to application requirements. Particular functions, which the programming service provides,

are functions for coordinated installation, upgrade and removal of software components or data files, and querying the state of software components on a sensor node.

VI. SUMMARY AND OUTLOOK

A flexible protocol architecture for WSNs has been introduced, which is well-suited for capturing the context surrounding service users in order to enable a variety of advanced context-aware applications in beyond 3G systems. To obtain an efficient cross-layer optimized protocol design, the system architecture has been divided into three modular, easily re-configurable subsystems, namely the connectivity, middleware, and management subsystem. In this paper, we have listed and discussed all service functions provided by each subsystem. More details on the protocol functions including a formal subsystem service/interface specification of the protocol stack can be found in [8].

The protocol stack architecture has been developed during the first year of the IST project e-SENSE. During this initial phase, the work has mainly focused on system aspects related to the WSN. In order to integrate the WSN into a beyond 3G mobile communication system, the protocol stack architecture needs to be extended with additional service functions and corresponding service interfaces. These additional functions, however, are only required for sensor nodes that act as gateway to the beyond 3G communication system. Furthermore, mechanisms and service functions to support runtime re-configurability of the protocol stack will be added. Work in this area is already in progress and will be the centre of attention in the second year of the project.

REFERENCES

- [1] e-SENSE website, www.ist-e-sense.org
- [2] D. Culler, et al., "Towards a sensor network architecture: lowering the waistline," in HotOS X - 10th Workshop on Hot Topics in Operation Systems, Santa Fe, NM, US. USENIX, 2005.
- [3] J. Polastre et al., "A unifying link abstraction for wireless sensor networks," in Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys '05), San Diego, CA, USA. ACM, 2005.
- [4] P. J. Marron, A. Lachenmann, D. Minder, J. Haehner, R. Sauter, and K. Rothermel, "Tinycubus: A flexible and adaptive framework for sensor networks," in Proceedings of the 2nd European Workshop on Wireless Sensor Networks, Istanbul, Turkey. USENIX, Jan. 2005, pp. 278–289.
- [5] ZigBee Specification v1.0, April 2005.
- [6] "IEEE Std 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specification for Low-Rate Wireless Personal Area Networks," IEEE, October 2003.
- [7] C. Lombriser, D. Roggen, M. Stäger, and G. Tröster, "Titan: A Tiny Task Network for Dynamically Reconfigurable Heterogeneous Sensor Networks," Accepted for Publication at the Symposium on Communication in Distributed Systems (KiVS) 2007, February 2007.
- [8] e-SENSE Deliverable, "D.2.2.1 – Initial e-SENSE system architecture," download: <http://www.ist-esense.org>