Integration of Wireless Sensor Network Services into other Home and Industrial networks using Device Profile for Web Services (DPWS)

Ayman Sleman
Automation and Process Control Engineering,
University of Wuppertal,
Wuppertal, Germany
ayman.sleman@uni-wuppertal.de

Reinhard Moeller
Automation and Process Control Engineering
University of Wuppertal,
Wuppertal, Germany
r.moeller@computer.org

Abstract— Wireless sensor networks (WSN) have recently been proposed for a large range of applications in home and industrial automation, and for health and environment monitoring. Especially the ZigBee standard is getting rising attention as it can be used to implement wireless sensor networks, because of its low data rate and low power consumption. Increasingly there is a need to access wireless sensor network services from other IP-based networks. The DPWS has been chosen to implement the integration of the different forms of networking services. DPWS uses web services, xml, WSDL and SOAP protocol to connect various devices and services in home and industrial applications. It allows collaboration and dynamic reconfiguration of network services and devices. This paper will describe our idea of how DPWS can be used to integrate a wireless sensor network into other IP-based networks.


I. INTRODUCTION

Recently, different types of networks can be seen in every place in home, car, factories and companies. Furthermore, wireless sensor networks (WSN) are becoming more and more important for home and industrial applications, and the need to access these networks from other existing networks is continuously increasing. The benefit of WSN will be largest, if the exchange of data between WSN and other networks is bidirectional and happens in suitable time. In general SCADA systems are used to manage real-time data in industrial automation, where a gateway is used to exchange data between MODBUS/TCP and WSN and such gateways are available on the markets. In these solutions OPC and DCOM programming technology are used.

Our idea is to benefit from the success of web services in other distributed IT applications like SAP, ORACLE, which offer data exchange between clients and web services using J2EE or .Net and have achieved great success in connecting business applications across corporate networks and the Internet. The use of web services, WSDL and SOAP allows developers of distributed industrial and home applications to connect devices written in different programming languages and from different manufacturers with each others. The paper describes how DPWS can be used to provide a secure model to access a wireless sensor network from other IP-based networks.

II. WIRELESS SENSOR NETWORK

Wireless sensor networks (WSN) have recently been proposed for a large range of applications in home and industrial automation. It consists of many tiny nodes, which have several sensors and a radio interface that depends on the IEEE 802.15.4 standard that supports large number of embedded devices in one network. WSN can be used for many applications such as environment monitoring, medical applications, robotic systems and home and industrial automation. WSN uses ZigBee standard (IEEE 802.15.4), which is a standard for low-rate wireless personal area networks (LoWPAN).

This IEEE 802.15.4 standard defines two layers of the OSI model: the PHY (physical layer) and MAC (media accesses control layer). The others layers of OSI model are left to the developer.

The main features of a ZigBee standard device are:
- Low data rate (maximum 127bytes/s).
- Low power (usually uses 2 AA batteries for up to 2 years).
- Low cost.
- Uses three frequencies: 868, 915 MHz and 2.4 GHz.
- Low bandwidth (250 kbps in the 2.4 GHz band).
- Supports three network topologies (star, tree, mesh).
- Supports a large number of nodes in the network.
- Uses Ad hoc networking.
- Establishes connections quickly.
- Supports large numbers of network nodes.
- Supports Built-in AES-128 encryption and authentication.
III. SERVICE-ORIENTED ARCHITECTURE SOA

A service-oriented architecture (SOA) is a distributed software architecture that depends on web services for building systems. In this architecture the client looks up for the services that are normally registered in the services directory, and the data is usually exchanged in XML format. The web services architecture is based on several protocols (SOAP, WSDL and UDDI). The communication between the web services and the client is based on the SOAP protocol and uses XML as data format as shown in the Fig. 1.

**Fig. 1. Components and Protocols of SOA Architecture**

Fig. 1 shows the core web services architecture:
- **XML (eXtensible Markup Language)** is used to define the data formats of the messages that are sent to and received from services.
- **UDDI [4]** (Universal Description, Discovery and Integration) is central services directory. In this directory the client looks up for services.
- **WSDL [5]** (Web Services Description Language) has XML format and is used to describe the services and its binding to transport protocols.
- **SOAP [6]** (Simple Object Access Protocol) is used to transfer messages between clients and services. The services that use SOAP are independent of the system platform and programming language. Usually the messages transferred by SOAP are formatted regarding to WSDL definitions. Using SOAP over HTTP allows for easier communication behind proxies and firewalls.

IV. DEVICE PROFILE FOR WEB SERVICE

Device Profile for Web Services (DPWS) is a profile designed for embedded systems and devices with small resources. It is also called device-level protocol, and it is a new SOA protocol and is considered as a successor for UPnP (Universal Plug and Play) [12].

DPWS and UPnP are both platform-independent protocols, and there are small differences between them, indeed DPWS is designed as a base to upgrading from UPnP to UPnP Version 2.

DPWS uses a set of web service protocols such as IP, TCP, UDP, HTTP, SOAP, XML, WS-Discovery, WS-Eventing, WS-Transfer, WS-Security and WS-Policy, which enable the devices to dynamically be discovered and described and also to exchange messages, send and receive data and events. DPWS uses XML schema to describe the services and to exchange data.

**Fig. 2 describes the stack of DPWS protocols.**

<table>
<thead>
<tr>
<th>Application Protocols</th>
<th>WS-Discussion</th>
<th>WS-Eventing</th>
<th>WS-Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP</td>
<td>HTTP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCP</td>
<td>IPv4, IPv6, IP multicast</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**The core Protocols of the DPWS stack are as follows:**
- **WS-Addressing [7]** uses SOAP, where all addressing information of a message will be put into the envelope of the SOAP message header, which can be extracted by one of the transport layer protocols.
- **WS-Discovery [8]** is used to dynamically discover network devices using plug-and-play, ad hoc protocol. It uses a multicast protocol for searching and locating of network devices.
- **WS-Policy [9]** determines the policies of web services. WS-Policy assertions express the capabilities and constraints of a particular Web service.
- **WS-Metadata Exchange** allows to dynamically obtaining web service metadata (WSDL, XML and policy).
- **WS-Security [10]** provides a set of rules to ensure a secure SOAP message exchange through message integrity, message confidentiality, and single message authentication.
- **WS-Eventing [11]** allows a web service (event consumer) to subscribe to another web service (event source) in order to receive event notification messages.
V. IPv6 /6LoWPAN MAPPING

There is a need to access the wireless sensor networks from other IP-based networks. In order to achieve this need the TCP/IP packets should be transferred through the ZigBee packets, but there is a difference between the OSI model of TCP/IP and the OSI model of ZigBee, this resulting into a need for a new layer that makes the adaptation between the two standards as shown in Fig. 3.

![Figure 3. IP based Wireless Sensor Node](image)

This adaptation layer is defined by the 6LoWPAN work group, which is responsible for standardization efforts of 6LoWPAN, and concerns about IPv6 over IEEE 802.15.4 [12].

- **IPv6 (Internet Protocol Version 6):** is a transport protocol located in the third layer of the OSI Model. It is the successor to IPv4. The most significant change from IPv4 to IPv6 is the size of IP addresses. IPv6 addresses size is 128 bits instead of 32 bits in IPv4. IPv6 addresses consist of two logical parts, a 64-bit network prefix, and a 64-bit host-addressing part, which is often automatically generated from the interface MAC address. IPv6 address consists of 8 groups of 16-bit hexadecimal values separated by colons (:-).

IPv6 addresses are classified in three types:

- Unicast Address is applied to one network interface and is delivered to the host with the specified address.
- Multicast Address applied for multiple network interfaces, and the packet is delivered to all hosts with the same address.
- Anycast Address applied for multiple network interfaces, but the actual packet is delivered to one of them.

- **6LoWPAN:** The WSN uses the standard IEEE 802.15.4 that supports a maximum frame size of 128 bytes. In order to move the IPv6 packet over the ZigBee network the IPv6 requires a support of packet sizes larger than the 128 bytes. The 6LoWPAN standard defines an adaptation layer for mapping from the IPv6 packets to the 6LoWPAN. This layer is responsible for the fragmentation, reassembly of fragments, IPv6 header compression and handling of the mesh addressing and broadcast headers. The 6LoWPAN supports two types of IEEE 802.15.4 addresses, the extended addresses (IEEE 64-bit) and the short addresses (16-bit). The IEEE802.15.4 standard defines also four types of frames:
  - Data Frame
  - Beacon Frame
  - MAC Command Frame
  - Acknowledgement Frame

Usually the IPv6 packets are transferred using the data frame only.

- **IPv6/6LoWPAN mapping:** This mapping is implemented through the adaptation layer. The key functions of the adaptation layer are the header compression and fragmentation. Usually the IPv6 header can be compressed by ignoring some fields that can be obtained from the source and destination addresses of the IPv6 packets that are specified in the header of 802.15.4 data frames. The fragmentation of IPv6 packets into multiple link-level frames is required to accommodate the IPv6 requirement of minimum MTU (maximum transmission unit) of link layer of 1280 bytes.

Each node can generate a link local IPv6 Unicast address from its extended EUID64-bit address of form FE80::EUID64/10 using stateless auto-configuration [13]. That means the extended EUID64-bit addresses can be used to obtain an IPv6 interface identifier and to connect directly with a 802.15.4 node of LoWPAN. It is also possible to use the 16-bit short addresses for stateless auto-configuration. Usually the short addresses will be available only after the association with the coordinator, and they are unique within the LoWPAN.

The coordinator coordinates the LoWPAN and each LoWPAN has one PAN coordinator with PAN_ID, the other nodes in the LoWPAN communicate with each other using the PAN_ID.

The 16-bit short addresses can be used to obtain an IPv6 interface identifier using the auto-configuring method. In this case a specific 48-bit pseudo-MAC address can be obtained by concatenating the 16-bits PAN ID, 16 zero bits and the short address (from left to right). The 48-bit pseudo-MAC address would be as shown in Table 1:
TABLE I. 48- PSEUDO-MAC ADDRESS

<table>
<thead>
<tr>
<th>48- Pseudo-MAC address</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAN ID</td>
</tr>
<tr>
<td>16 bits</td>
</tr>
</tbody>
</table>

The IPv6 interface identifier can be then obtained from this 48-bit pseudo-MAC address by inserting the hexadecimal value 0xFFFE between the third and fourth bytes of the 48-bit pseudo-MAC address.

The IPv6 interface identifier is shown in Table 2:

TABLE II. 64- IPV6 INTERFACE IDENTIFIER

<table>
<thead>
<tr>
<th>64-IPv6 Interface Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAN ID</td>
</tr>
<tr>
<td>16 bits</td>
</tr>
</tbody>
</table>

Additionally the universal /local bit, the second low-order bit of the first byte of the resulting IPv6 interface identifier should be 1 to indicate that this identifier is locally administered.

VI. DPWS GATEWAY SOLUTION

Our solution of DPWS gateway comprises a model of web services that uses XML, WSDL and SOAP and enables the exchange of data between WSN and other IP-based networks. Fig. 4 shows the overall concept of a DPWS gateway, which allows applications to read data from the WSN sensors and to issue commands to the WSN devices. It is running on a PC under Windows or Linux. The PC running a DPWS gateway has access to an IP network and it connects to at least one coordinating device in WSN which acts as a bridge between other WSN devices and the PC.

A. The components of the systems

- **Client:** Each client may be an application that uses any of TCP, UDP or HTTP that in turn are based on IP protocol. The IP packets will be transferred to the coordinator that converts the packets into 6LoWPAN format and sends them to a specific sensor node. The client can issue a request to a specific node, receive responses and events from nodes, obtain service descriptions and use the services of nodes.

- **DPWS Gateway:** It intermediates between the client and the wireless sensor network and manages the traffic on the network. The gateway registers the sensor nodes in a routing table, obtains their WSDL file, requests services from the wireless sensor nodes and offers them to the client, collects and forwards the events, the metadata and sensor data between the client and the nodes.

- **Wireless Sensor Nodes:** Each node has a unique EUID-64, it can join the LoWPAN and advertise its services using WSDL file. The node can provide its data using get-method of its services or can perform actions using set-method of its services. The node is also able to send and receive events.

- **Coordinator:** It is a ZigBee node connected to the gateway using USB or serial port. Through this node the gateway can connect with WSN. It takes the packets from wireless sensor network and encapsulates them into IPv6 Packets, or takes the packets from the client through the Ethernet interface and translates it from IPv6 into 6LoWPAN form, which can be understood by WSN.

B. The functionality of the system

The interaction between client and wireless sensor network can be done in a specific sequence as shown in Fig. 5.

The interaction can be arranged in the following stages:

- **Addressing:** The client and DPWS gateway have specific IP addresses. Each sensor node also has a unique EUID-64. When the wireless sensor is powered on, it sends its EUID to the DPWS gateway that in turn registers the EUID in a routing table. After that the wireless sensor is part of the LoWPAN.

- **Advertising and discovery of services:** Once the wireless sensor node has become part of the LoWPAN, it sends a WSDL file containing the description of its services. That means each node informs all other network members of its services, and also it can be informed about the presence of new members. When a new client joins the network, it sends a search request for other network members in order to get information about their advertised services. Usually, search and discovery of services are done using multicast method.

- **Getting a service's description:** Once the client has discovered the wireless sensor nodes it needs to obtain more information about the services of each node. The
client sends a request to the DPWS gateway in order to get metadata about a specific node. The DPWS gateway gets the metadata information from the node and sends it back to the client. For each service advertised by the node, the service description defines the actions to which the service responds and the corresponding functions the node offers and the format of the messages. Usually the metadata, which describe the node’s capabilities, is presented by a WSDL file using xml format.

- **Using node services (Get and Set functions):** Once the client receives the metadata it knows the functions and possible actions of the node. Then it can get data from the node or send actions to it. To request an action on a node’s service, the client sends a control message to the node. The node performs then the action and returns a feedback to the client.

- **Asynchronous Events:** The client can subscribe to use the asynchronous events of the nodes. Usually the node sends an event when its state changes from one state to another, to inform the client about its new state.

![Sequence Diagram](image)

**Figure 5. DPWS gateway sequence diagram**

**CONCLUSION**

The presented paper has discussed the need and how to integrate wireless sensor networks into other existing IP-based networks.

Using the 6LoWPAN it is possible to connect a wireless sensor network with the internet and other IP-based networks in home and industrial environments. The 6LoWPAN also implements the header compression and fragmentation as well as reassembly of fragments in order to map from IPv6 to ZigBee network.

The DPWS Profile was chosen to achieve the implementation of a DPWS gateway because it is suitable for embedded devices in home and industrial automation, and because of its interoperability and its automatic networking and discovery of services.

**REFERENCES**

[5] http://www.w3.org/TR/wsdl (Web Services Description Language (WSDL)).