Internet of Things Application in WSNs

Mirko Kosanovic[1], Milos Kosanovic[2]
[1]Nis, Aleksandra Medvedeva 20
18000 Nis, Serbia
{milos.kosanovic@vtsnis.edu.rs, mirko.kosanovic@vtsnis.edu.rs}

ABSTRACT: *In recent years, we are witnessing an increasing presence of a large number of different sensors that supply us with different data. The Wireless Sensor Network (WSN) has become a basic technology that has enable us to collect these data and that almost all complex phenomena in the physical world convert a simple set of units and zeros. Further evolution of this technology into Internet of Things (IoT) expands the possibilities of much application. This paper presents some of the basic approaches and technologies that are used today. At the end of the work, a typical solution for IoT application is proposed.*

Keywords: Wireless Sensor Node, Internet of Things, Cloud Computing, Mobile Application

Received: 9 April 2020, Revised 30 July 2020, Accepted 16 August 2020P

DOI: 10.6025/jism/2020/10/4/127-133

1. Introduction

WSN increasingly have an impact in our everyday lives. They reveal a wide range of applications across different domains, including health, assistance and improved living conditions, industrial and production monitoring, traffic and transport controls and many other areas. The dream of intelligent devices that independently perform many complex tasks became a java because it was realized by the implementation of the IoT concept. The IoT makes it possible to connect WSN into entire Internet infrastructure through various communication and information technologies. Already in 2008, the number of Internet connected objects on our planet has exceeded the total number of people living on it. At the moment, there are over 10 billion interconnected facilities, and according to Cisco Systems, Dave Evans futurists, it is expected that at the end of 2020 this figure will reach about 50 billion interconnected computer objects [1]. His vision is that in the future, the world will be full of networked, smart devices that will have unique identification (IP number or radio frequency identification - RF number). All of these devices will be equipped with a large number of different sensors and will be able to completely independently collect a large number of different data that will be exchanged or forwarded to the parent device (sink), which will enable them to be available on the Internet by wireless connection. It has emerged as the ultimate product of the three core technologies that in recent years have undergone the greatest progress and development: wireless communications, micro electro mechanical systems (MESN) and Internet technologies. One of the main advantages of IoT is that it allows a large number of people to create very powerful and cheap applications with very little knowledge and experience. All this is made possible by the development of a large number of micro-platforms, such as Arduino, Propeller and Microchip PIC families, and their integration through wireless communication with the Internet. In other words, this technique requires engineers to equally share knowledge of hardware and software. This is necessary in order to apply this knowledge in a coherent and integrated way in order to obtain an applicable IoT application [2, 3]. On the other hand, there has been a major development of Internet technologies that have contributed to realizing a large amount of data from a large number of computers, and this is almost always presented to clients. First of all it refers to new generation of distributed computing technologies: Cluster computing, Grid computing, Big Data and Cloud computing. They were now a precondition for developing and using new applications, WEB service oriented software. These technologies also have the ability to use messaging mechanisms such as Email, SMS or to exchange messages via social networks and blogs [4]. All of these features were ideal for their benefits to be utilized to connect to a single distributed source of a large number of data, which are WSN.

The rest of this paper is organized as follows: Section 2 provides the actual open issues of development of IoT. The advantages of 6LoWPAN technology are explained in Section 3. Section 4 discusses about the CoAP application protocols and its applicability into IoT applications. The types of web services that are applicable in IoT applications are discussed in Section 5. In Section 6 we present one proposal for an construction of standard IoT application. At the end, Section 7 concludes this paper by looking at the future research and recommendations which are required to make the IoT application more effective.

2. Research Questions

Following the IoT vision, WSN has made significant progress in recent years and has become closer to the structure and organization of standard protocols from the TCP/IP suite. Traditional WSN has emerged from simple, isolated systems for the monitoring natural phenomena into interoperable, powerful systems that are connected to the Internet. Cloud computing and IoT, which are two distinct technologies, are expected to become an integral part of the Future Internet considering the rate at which they are being adopted and used. This development has taken place in several different fields, of which they are particularly distinguished:

2.1. Protocol Customization

New customizable protocols have been developed that could be implemented on modest resources available to sensors that were compatible with standard protocols from a TCP/IP suite. 6LoWPAN and CoAP are typical protocols that have enabled WSN to become more powerful in terms of interoperability and scalability.

2.2. Powerful WSN Application

Today sensor nodes(SN) have many multimedia sensors such as camera, microphone, radio frequency identification reader (RFID reader), radio frequency identification TAG (RFID -TAG)). Such progress has increased the interoperability and strength of WSNs and opens one new array: Big Data. It's a term that describes the large volume of data – both structured and unstructured – that inundates a business on a day-to-day basis. But it's not important the amount of data, than what someone do with these data. Big data can be analyzed for insights that lead to prediction of events, better decisions or strategic moves.

2.3. Advancement in Hardware Technologies

Today SNs have significantly more powerful resources and computing capabilities: from 8b to 32b microcontroller, from 8MHz to 80MHz, from 10KB to 128KB RAM, from 48KB to 512KB flash memory. On other side we have cheaper hardware (Arduino) and an increasing number of different Wi-Fi modules(ESP2866) that connected SN with Internet.

2.4. Development of New Operating Systems

Another requirement to consider is the ability to remotely change the services (program code) provided by the SN during its lifecycle. This possibility enabled the use of many advanced techniques like Web services or possibility of use Web Socket protocols, which has enabled many real-time applications.

3. 6LOWPAN Technology

6LoWPAN is an open standard defined in RFC 6282 by the Internet Engineering Task Force (IETF). Initially, 6LoWPAN was designed to support the IEEE 802.15.4 standard for lowpower wireless networks in the 2.4-GHz area. Now it become a standard that has been adapted and can be used in a variety of other networking technologies, such as the Sub-1GHz low - power RF, Bluetooth Smart, power line control (PLC) and low-power Wi-Fi. It is based on standard IEEE 802.15.4 (LoWPAN) that is applied to devices that have very limited resources such as wireless SNs. One of the basic prerequisites for each SN is the minimum power consumption in order to achieve the maximum life expectancy [5].

A lot of different technologies have been developed for wireless communication between SNs that allow access to the Internet, but it is quite clear that technologies based on the TCP/IP protocol suite have an advantage [6]. As most protocols from the TCP/IP suite represent very demanding protocols, with only headers being large enough (IPv4-24 bytes, IPv6-40 bytes, UDP-8 bytes, TCP-24 bytes), it is clear that the full implementation of these protocols in the WSN was a complete failure, from the point of energy efficiency and efficiency. For only 2-3 payload data, which is usually the size of the data transmitted through WSN, we would have to transmit 30 bytes, and therefore we spend a large amount of electricity, which indirectly means that life of these SNs was reduced to only a few days [7]. Technology 6LoWPAN put this requirement as the primary goal that must be fulfilled.

6LoWPAN network connects to other IP networks through one or more edge routers whose task is to transfer IP packets between different networks (see Figure 1). Connecting to other IP networks can be provided through various links-links such as Ethernet, Wi-Fi or 3G/4G. As the 6LoWPAN standard defines IPv6 protocol operations over the IEEE 802.15.4 standard, routers must also support an IPv6 transition mechanism that allows the connection of the 6LoWPAN network to IPv4 networks, as defined in the RFC 6146. This frees SN in the 6LoWPAN network to have this mechanism implemented, while at the same time allowing them to interact with IPv4 networks without interruption. Generally speaking, the basic goal of each 6LoWPAN network is to enable the use of standard Web services, such as REST, XML, JSON, and other devices with very limited resources, such as SNs, in environments that have a very high percentage badly received messages, LLNs (Low-Power and Lossy Networks) networks.

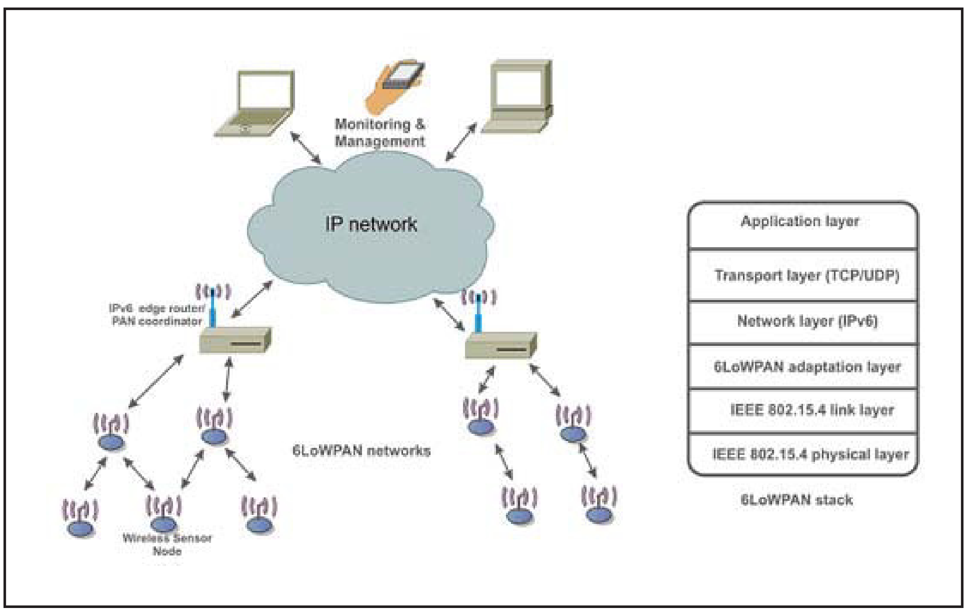

Figure 1. Typically connect devices to a 6LoWPAN network

4. CoAP Application Protocol

The integration of the REST architecture within WSN is not at all a simple task from the simple fact that this is a very limited resource available to the SN within that network. Typical SNs are powered by a battery, a strictly limited source of power, and have several kB memory and CPU that have limited computing capabilities. Consequently, any direct application of the original protocols from the TCP/IP suite is completely inapplicable because in this way the lifetime of all SNs in the network would be very short [8]. That is a main reason to develop a new special protocol, by the IETF working group, called the CoAP (Constrained Application Protocol), with the goal of being the basic Web transport protocol that would be applied to WSN and which would replace the HTTP protocol. CoAP seeks to apply the same data transfer system as the HTTP, but with significantly less resource requirements. It supports one part of the HTTP function, but also extends this set with its own functions to simplify communication between two SN, i.e. enabled M2M (Machine to Machine) communication. In this way, the services offered by Web services are gaining importance because now each SN can use them and participate in their expansion [9].

The core intention of the CoAP protocol is to provide a generic Web protocol that the SNs can communicate with. It is very similar to HTTP protocol (Fig.2), but its goal is not simply to compress HTTP packets, but to implement a reduced set of system messages that will enable M2M communication. To this end, CoAP uses the following four messages: CON (Confirmable), NON (Non-Confirmable), ACK (Acknowledgment) and RST (Reset). The operating mechanism of this protocol is similar to the client-server communication model because the client always requires some service from the server. After the requested request, the server responds including the unique message ID received with the request (see Figure 3). By looking at Figures 2 and 3, it can be seen how much the CoAP protocol is used up because it only uses two packets in normal communication, as opposed to an HTTP protocol that requires 7 packets. A particular advantage of the CoAP protocol is that minimum resources are needed that each SN needs to be able to be implemented: 10 kB RAM and 100 kB of memory space for program code (RFC 7228) [10].

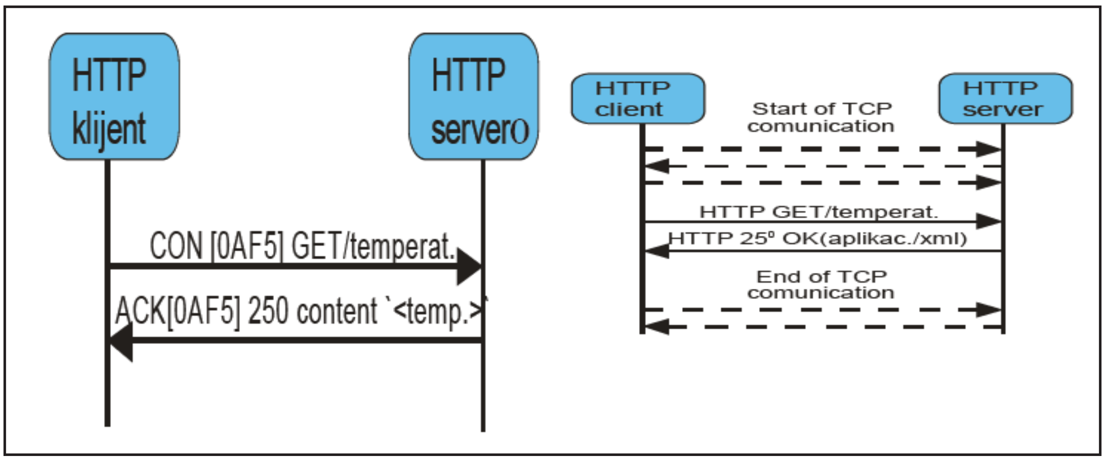

Figure 2. HTTP and CoAP comunication

5. Web Services

Web services represent a new look towards writing applications that are applied on the Internet. Their goal is to enable the connection of various information systems installed in distributed environments. They represent one type of distributed application that consists of several application functions that can be programmed from anywhere that has the ability to connect to the Internet. In doing so, the consumer and the provider of these applications use the messaging system to exchange their requests and answers, most often through the HTTP, whilst that communication does not depend of the resources which a provider and a consumer of these services have. In other words, the Web service is not, at all, tied to a specific hardware and software platform used by both parties in communication. It's just enough to support some of the protocols for sharing information such as HTTP or SMTP protocols. A typical architecture in which Web services are used (see Fig. 4) is a client-server architecture that allows one network component to play the role of a service provider, service user or service broker. With their simplicity and great capabilities, they have today become an integral part of almost all modern information systems.

The very rapid development of the Internet as well as related technologies led to the development of many technologies that were applied to Web services that were changing rapidly. From today's point of view, two Web development platforms have emerged as the leading: Microsoft.NET and Sun J2EE. Regardless of the chosen platform, all Web services can easily call each other. In general, we can split all Web services into two large groups: Big Web Services and RESTful Web Services (see Figure 4).

"Big" web services are based on the SOAP protocol for easy access to the sites and often contain a WSDL language for describing Web services. WSDL enables us to call each Web service when it is called, and which contains this function, it can describe itself, the operations it supports how we can use them. The details that WDSL provides can be messages, operations, links, and Web service locations. A SOAP message is usually an XML document defined by an XML schema. In the SOAP Web service, SOAP messages are not the subject of a programmer's interest in creating a Web service or writing a client code for using the Web service, but they only enable the communication of a client application that calls the Web service and the Web service itself. This type of Web service is suitable when we need asynchronous processing, reliability and pre-formulated State-full operations. If the application needs further information, SOAP provides an additional specification in the Web services structure to support the query (security, transactions, coordination, etc.) [11].

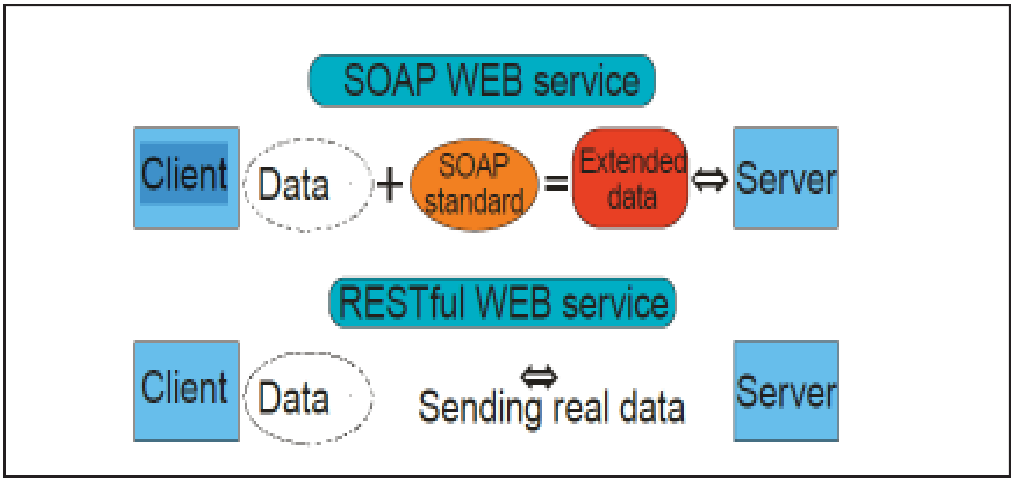

Figure 3. SOAP i RESTful Web services

RESTful web services are based on architecture called REST - Representational State Transfer. REST is not dependent on any protocol, but almost every RESTful service uses HTTP as the basic protocol. These services are much better integrated with HTTP than SOAP services, and as such do not require XML, SOAP messages, or WSDL definitions. The introduction of REST Web services into an attempt is to overcome the complexity of SOAP Web services. Basically, there is a big difference between these two types of Web services and does not exist so that the SOAP Web service using the HTTP protocol for messaging is one specific case of the REST Web service[11].

6. Proposed System and Architecture

In order to address the all mentioned issues of flexibility and functionality of IoT in this paper, we present one suggestion for standard IoT application through a layered structure of IoT. Each layer in this structure has its own defined task to be performed. The basic architecture of a IoT based application can be considered through four different layers (see Fig.4):

6.1. Perception Layer
Consists of different connected physical sensors which are supposed to be used for monitoring events in our application.

6.2. Interface Layer
Represent control units which are collecting data from the perception layer. This unit mainly consists of different tiny, inexpensive wireless sensor nodes which are network organized and are deployed over a wide geographical area, capable to integrate continuous and unobtrusive measurement, computing and wireless communication, completely autonomously.

6.3. Network Layer
The processing unit leads the accumulated data into the Network Layer, which provides two way communication and connectivity pathway via Cloud or WEB infrastructure, after which the data finally reaches the Application Layer and end user.

6.4. Application Layer
The Application Layer via which control and monitoring of the appliances, installed at mobile, lap-top or tablet, takes place from anywhere in the world.

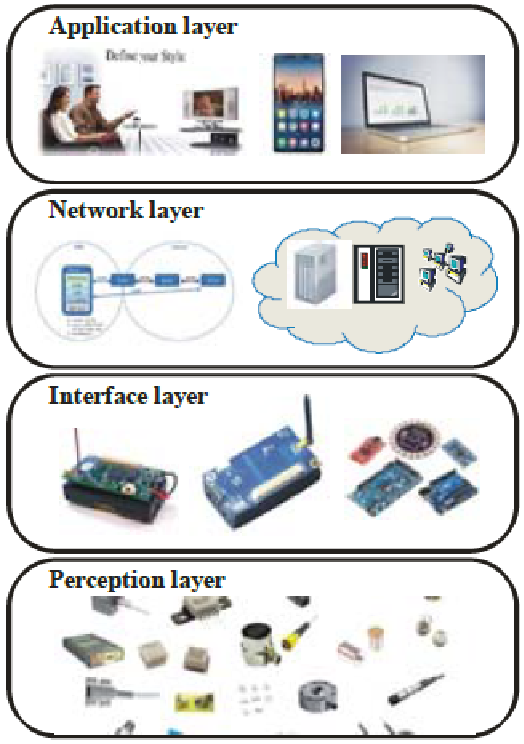

Figure 4. Layer architecture of IoT application

The architecture presented in this paper can be customized in different ways in order to accommodate different application scenarios with minimum recoding and design. This system allows authorized home owners to remotely control and monitor events via smart phone application with graphical user interface (GUI). The proposed system represents a classical client-server application based on Arduino microcontroller system, Wi-Fi (ESP2866 shield), ThingSpeak platform and the Android compatible mobile application. The ESP8266 WiFi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your home WiFi network. It offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor. This shield has the capability to be used both, as a client or a server. Since Arduino already supports a TCP/IP stack, we have focused on implementing software to connect it to the remote user. Software of the proposed IoT application is divided into three main parts: client application on smart mobile, server software application on some Web or Cloud devices and microcontroller firmware. The client application for Android system can be done using Google Android Programming IDE - App Inventor and Java programming language. The client application communicates with the microcontroller through ThingSpeak cloud. ThingSpeak is an open-source IoT platform that lets you collect and store sensor data in the cloud and develop IoT applications. This platform has its own API to store and retrieve data from sensor nodes using the HTTP protocol and let you to analyze and visualize your data in MATLAB, and then act on the data. Sensor data can be sent to ThingSpeak from different microcontroller system such as: Arduino, Raspberry, BeagleBone Black or other hardware[12]. Using the REST API request methods such as GET, POST, PUT, and DELETE, we can create a channel and update its feed, update an existing channel, clear a channel feed, and delete a channel. We sent a Java Script Object Notation (JSON) GET Request to ThingSpeak by using REST API Web Service and channel ID and field number within its parameters. We received the response from ThingSpeak in JSON format and populated the tables in the Android application by using JSON Parser.

7. Conclusion

In this paper we discuss current technologies that enable us to take advantage of all IoT applications. The fact that we can present a huge amount of collected data from different sources to a large number of clients, provides unprecedented opportunities for the development of a huge number of applications. On the other hand, the simplicity of building an IoT application is simply fascinating. We plan further research to be in the direction of implementing a CoAP protocol into IoT application as well as the benefits that can be gained from considering large amounts of data – Big Data analytics.

References

[1] Evans, D. (2019). The Internet of Things [INFOGRAPHIC], http://blogs.cisco.com/diversity/the-internet-of-thingsinfographic/pos. 25.02.2019

[2] Ian G. Smith (2012). *The Internet of Things 2012 New Horizons*, Halifax, UK, 2012

[3] Gubbia, J., Buyyab, R., Marusic, S., Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions, *Future Generation Computer SysteSN*, 29, 2013, p 1645–1660

[4] Adrian McEwen, Hakim Cassimally. (2014). *Designing the Internet of Things*, John Wiley and Sons.

[5] Raghunathan, V., Ganerival, S., Srivastava, M. (2006). Emerging techniques for long lived wireless sensor networks, *IEEE Communication Magazine*, 44 (4), 2006

[6] Beraka, M., Mathkour, H., Gannouni, S., Hashimi, H. (2012). Applications of Different Web Service Composition Standards, *IEEE International Conference on Cloud and Service Computing (CSC)*, p 56-63, Nov. 2012

[7] Kosanovic, M., Stojcev, M. (2011). Connecting Wireless Sensor Networks to Internet, Facta Universitatis, *Series: Mechanical Engineering*, 9 (2), 2011, p 169-182

[8] Rellermeyer, J., at all. (2008). The Software Fabric for the Internet of Things, *In*: Proceedings of the First International Conference on the Internet of Things, Zurich, 2008.

[9] Guinard, D., Trifa, V., Pham, T., Liechti, O. (2009). Towards Physical Mashups in the Web of Things, *In*: Proceedings of INSS 2009, Pittsburgh, USA, 2009.

[10] Shelby, Z. (2014). Hartke, K., Bormann, C. (2014). The Constrained Application Protocol (CoAP); RFC 7252; The Internet Engineering Task Force: Reston, VA, USA, June 2014.

[11] Leonard Richardson., Sam Ruby. (2007). RESTful Web Services, O'Reilly Media, Inc., May 2007.

[12] Marco Schwartz. (2016). Internet of Things with Arduino Cookbook, Packt Publishing, 2016.