CoAP protocol for Web-based monitoring in IoT healthcare applications

Dejana Ugrenovic, Gordana Gardasevic, *Member, IEEE*

***Abstract* — Internet of Things (IoT) for healthcare applications is an emerging research field that has gained a lot of attention in the last few years. The paper presents the IoT remote healthcare monitoring system that provides the patient's conditions through Web browser. For simulation purposes, we use Contiki OS with 6LoWPAN protocol stack, and Cooja, the built-in Contiki simulator. CoAP is selected as application level protocol for remote data access and representation.**

***Keywords* — Internet of Things (IoT), Constrained Application Protocol (CoAP), Contiki, Healthcare.**

I. Introduction

Internet of Things (IoT) represents the concept of connecting various objects (physical or virtual) through wireless or wired connections, thus enabling them to communicate with each other as well as with outside world. The objects' addressing is one of crucial requirements in order to make them remotely accessible at any time and at any place [1].

The important parts of IoT architecture are Wireless Sensor Networks (WSNs). WSNs are composed of low-power sensors with low processing capabilities and constrained resources. The main sensor's task is collecting and sending data through the gateway to the outside world [2]. There is an important role of WSNs for IoT healthcare applications. Medical and healthcare systems gain benefits from WSNs that provide useful applications such as patient monitoring in real time, drug administration, help in diagnostics, tracking system of patient inside the hospital, etc. Remote monitoring of patient's condition is particularly helpful for doctors if the patient is outside the hospital [3],[4].

The data-link protocol, commonly used in WSNs, is IEEE 802.15.4. It is optimized for low-power devices with energy and memory constraints. The commonly used protocol for sensor's addressing in WSNs is IPv6, due to more address space and modular design. IPv6 protocol also provides the interoperability with other devices and enables Internet access for sensors and other IoT devices.

Dejana Ugrenovic is with the Faculty of Electrical Engineering, University of Banja Luka, Patre 5, 78000 Banjaluka, BiH (e-mail: ugrenovic.dejana@gmail.com).

Gordana Gardasevic is with the Faculty of Electrical Engineering, University of Banja Luka, Patre 5, 78000 Banjaluka, BiH (e-mail: gordana.gardasevic@etfbl.net).

Some adaptation mechanisms are required in order to enable IPv6 packets to be carried through the IEEE 802.15.4 link. 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Network) is the protocol that provides the header compression and fragmentation of IPv6 packets. It creates the adaptation layer over the IEEE 802.15.4 layer and enables wireless communication in networks with limited power [5].

Due to very high packet losses and communication unreliability in WSNs, as well as the constrained bandwidth and energy resources, routing protocols in classic networks are not adequate for WSNs. IEEE ROLL (Routing Over Low Power and Lossy network) group proposed the RPL (Routing Protocol over Low Power and Lossy network). RPL is a distance-vector protocol, where one sensor is set to be root node and it creates a tree-like structure that results in Destination-Oriented Directed Acyclic Graph (DODAG). Each node selects its preferred parent based on Objective Function (OF). The OF can be established on various metrics, such as hop count, ETX (Expected Transmission Count), throughput, etc. [6].

The REST (Representational State Transfer) architecture provides that IoT applications are developed on the top of Web services. REST is commonly used with HTTP (Hypertext Transfer Protocol) protocol, but HTTP is not suitable for constrained wireless networks due to code space implementation and network and energy resource usage. The Constrained Application Protocol (CoAP) is specified by Constrained RESTful environments (CoRE) working group in order to overcome some issues related to HTTP [7],[8].

The aim of this paper is to analyze and simulate the IoT healthcare system that monitors patient's health condition, and to access the relevant status parameters through Web browser at any time and any place. This will provide more secure healthcare applications and appropriate on-time reactions and measures. Patients that are not in critical condition may be moved from hospital, and the monitoring of their vital parameters in real time can be performed from their homes.

The remainder of this paper is structured as follows. Section II reports some related work in the field of IoT healthcare. Section III describes the CoAP protocol in more details. Section IV presents the simulation setup and results. Concluding remarks are provided in Section V.

978-1-5090-0055-5/15/$31.00 ©2015 IEEE

II. RELATED WORK

The IoT has a great potential in the field of medical and healthcare applications. Some of IoT-related technologies, such as body area sensors, advanced healthcare systems, wearable sensors, cloud-based platform for wireless transfer, storage, and display of clinical data, etc., are of particular interest.

Paper [9] describes the implementation of remote medical care system based on Android platform and ZigBee technology. Different physiological signals (ECG, blood pressure, etc.) are measured, collected and analyzed. The obtained results are sent through ZigBee wireless device to the central monitoring system. In [3], the system architecture with 6LoWPAN sensors is presented and analyzed. Such sensors are creating the Wireless Body Area Network (WBAN). Simulations are performed in Cooja, and the sensor platform used for simulation purposes is Zolertia. Two scenarios are presented, where the Packet Error Rate and throughput are selected as performance metrics.

The communication platform for healthcare system based on Contiki 6LoWPAN is presented in [10]. The platform used for analysis is Z1 from Zolertia, compatible with Tmote Sky. In this system, three nodes are sending data in real-time to the border router, which enables monitoring of patient's conditions (ECG, temperature, and acceleration). The PC is connected to the border router and it performs the function of bidirectional gateway between 6LoWPAN and the Internet. Paper [11] presents a survey of IoT common technologies used for the implementation of healthcare monitoring system. This system is using certain sensor properties as resources, while CoAP methods with REST architecture are used for resource management.

III. COAP PROTOCOL

CoAP protocol uses the REST architecture, where resources are identified by Universal Resource Identifier (URI). HTTP functionalities are redesigned for CoAP, but the main principles remain the same. The key difference between HTTP and CoAP is related to underlying transport layer. HTTP relies on connection-oriented TCP protocol, while CoAP is built on top of simple connectionless UDP protocol [12]. Fig. 1. shows the client-server interaction model that is used in CoAP. Clients are sending requests for specific action on resource of the server, and server, after processing the request, is sending the response code and the resource representation.

Fig. 1. Client-server communication in CoAP protocol.

The action requested by client is defined by method, URI, and in some cases meta-data about the request. There are four different methodes defined in CoAP standard:

- GET - for retrieving resource representation,
- POST - for transferring information representation,
- PUT - for updating resource on server,
- DELETE – for deleting resources [8].

The communication stack with CoAP is shown in Fig. 2. The PHY and MAC layer are based on IEEE 802.15.4 specifications, to enable low-power communications for wireless embedded devices in low-rate WPAN (Wireless Personal Area Network). The 6LoWPAN adaptation layer accommodates the transmission of IPv6 packet within the MAC frame. The network layer is represented by IPv6 mechanisms and RPL routing protocol designed for these types of networks. UDP protocol is commonly selected as transmission protocol for IoT applications, due to low overhead and simple mechanisms.

Application	CoAP
Transport	UDP
Network	IPv6/RPL
Adaptation	6LoWPAN
MAC	802.15.4
Physical	802.15.4

Fig. 2. CoAP communication stack

IV. SIMULATION SETUP AND RESULTS

Contiki is the lightweight operating system (OS), implemented in C programming language. It is purposely developed and created for the low-power devices in constrained environments [13]. Contiki has its own implementation of RPL protocol, named ContikiRPL [14].

Cooja is a cross-level simulator, built in Contiki OS. It provides the simulation on network level, OS level, and machine code level. Cooja has been developed in JAVA programming language, but the sensors that are simulated can be also programmed in C programming language. It allows simulation of sensors with non-Contiki OS, such as TinyOS [15].

We are using Cooja simulator for a simulation purpose. Contiki 2.7 is installed on Ubuntu 14.04 OS. The platform selected for simulation purposes is Tmote Sky. This platform is characterized by the following: 8 MHz, 16-bit RISC processor with TI MSP430F1611 10 kB RAM, 1 MB Flash memory and Chipcon CC2420 transceiver at 2.4 GHz with transmission range of about 100 meters [16]. Medium Access Control (MAC) and physical layer are defined by IEEE 802.15.4 standard. The mechanism used at MAC layer is CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). Radio Duty Cycle (RDC) mechanism selected for simulation purposes is the default built-in Contiki mechanism – ContikiMAC [17].

Table 1 describes sensor parameters relevant for the implementation of system shown in Fig. 3.

TABLE 1: SENSOR PARAMETERS

Sensor ID	Role	Data	Payload (Bytes)
ID1	Border router		
ID2	Router		
ID3	Server	Temperature	4 B
ID4	Server	ECG	40 B
ID5	Server	Oxygen saturation	5 B
D6	Server	Respiration rate	1 B
ID7	Server	Glucose level	3 B
ID8	Server	Blood pressure	2 B

The aim of this system is to enable the access to the patient's health status through Web browser. The communication is client-server, and in this case, Web browser acts as a client, and sensors placed on patient's body have a role of servers**.** Each server provides one particular resource, and each resource represents information about patient's health status. For example, server ID3 provides data about body temperature, while server ID4 provides information about ECG status.

The role of sensors in this simulation are described in Table 1. Each parameter has a different payload value, according to the specific requirements (some parameters, e.g. ECG signal, requires more data in order to be correctly analyzed). The Fig. 3. shows the Cooja simulation setup. For a successful communication between Web browser and CoAP client, the bridge between RPL 6LoWPAN network (network of sensors) and local IP network (client's network) is required. This is implemented within the border router functionalities, where Serial Socket Server option is enabled, thus creating the serial port on sensor (accessible through the port 60001). Tunslip tool is used on client side in order to connect the 6LoWPAN network with local network. The border router also has the root role in DODAG topology, while other servers actively participate in creating DODAG and routing packets. Servers are positioned outside the border router's transmission range, in which case the additional router (ID2) is required, in order to enable the communication between border router and servers. IPv6 addresses are assigned to the sensors according to their IDs in Cooja simulator. For example, the border router is marked as ID 1 and its IPv6 address is aaaa::212:7401:1:101, while the temperature sensor is marked as ID3 and its IPv6 address is aaaa::212:7403:3:303.

Two different implementations of CoAP protocol have been used in this approach. Erbium (Er) is a REST service for Contiki OS, developed by Swedish Institute of Computer Science [18]. Cooper (Cu) is an add-on for Mozilla Web browser [19]. Mozilla Firefox Web browser with Cu add-on represents a client that is accessing resources of CoAP servers (ID3,...,ID8) with GET method. The Fig. 4. shows the content of Web browser and presents information about body temperature of patient.

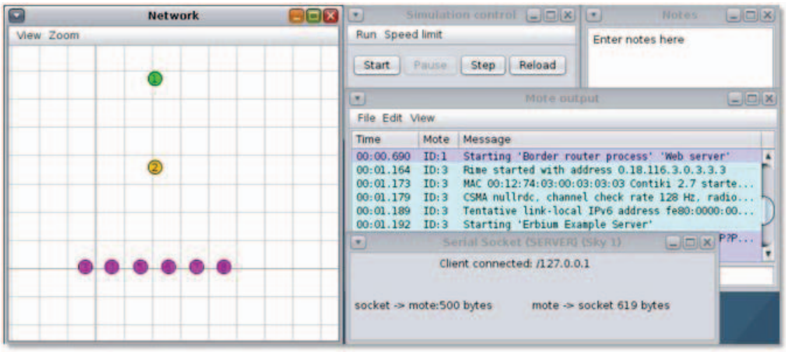

Fig. 3. Simulation setup and mote output in Cooja

Uniform Resource Identifier (URI) scheme is used for identification and locating of CoAP resources. The example of CoAP URI scheme is the following:

coap://[aaaa::212:7403:3:303]:5683/Temperature

The following list explains parts of URI scheme:

- coap – name of application protocol
- aaaa::212:7403:3:303 – IPv6 address of server
- 5683 – default UDP port for CoAP protocol
- Temperature – representation of resources

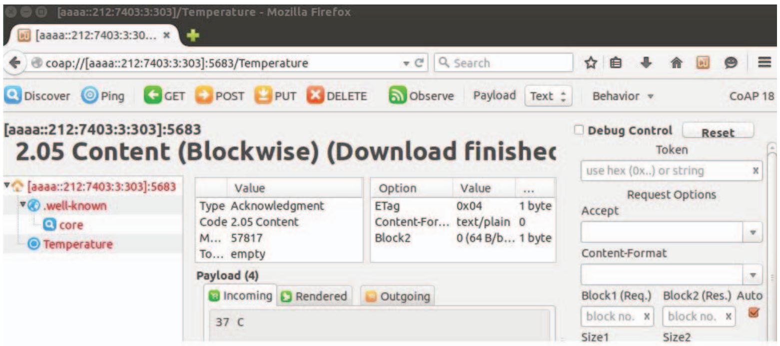

Fig. 4. The screenshot of Mozilla Firefox window

In order to analyze the packets' structure and flows, we have also used the Wireshark software for tracking packets on each level. Client's access server's resources with method GET is captured. The screenshot of Wireshark window, with the example of request/response message, is illustrated in Fig. 5.

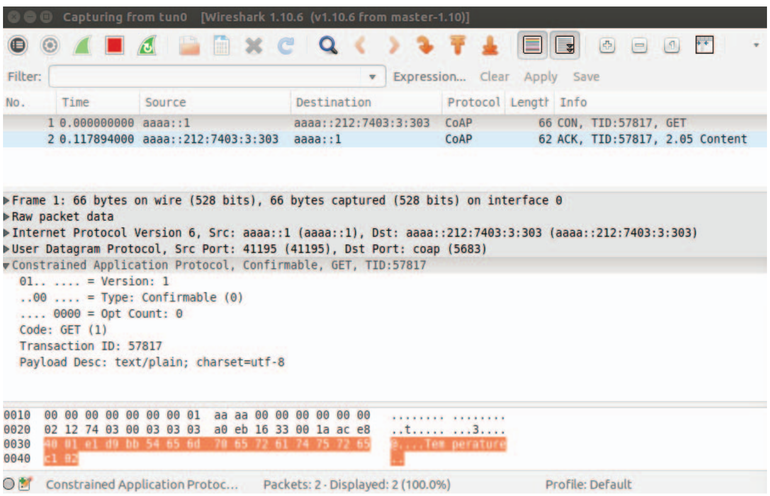

Fig. 5. The screenshot of Wireshark window

V. Conclusion

The IoT smart sensors for healthcare applications enable accurately measuring, monitoring and analyzing a variety of vital health status indicators, such as heart rate, ECG, blood pressure, levels of glucose or oxygen saturation in the blood, etc. These parameters are then collected and transferred to various IP end-devices. This approach enables, in a convenient way, a real-time access to the data of patient's health condition.

This paper presents the IoT remote healthcare monitoring system that provides the patient's vital parameters through Web browser. The paper also describes the required steps for connecting the 6LoWPAN network with the outside world and the Internet. The main focus of this work is the implementation of CoAP protocol in Mozilla Firefox Web browser, and the manipulation of resources by CoAP methods. The use of CoAP protocol for IoT applications overcomes some well-known issues related to resource-constrained environments. CoAP is reliable and effective application protocol, intended for use in WSNs. In the next steps of our research, the focus will be on security aspects of WSNs for IoT applications.

References

[1] O. Vermesan and P. Friess, Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems. Aalborg, Denmark: River Publishers, 2013.

[2] L. Mainetti, L. Patrono, A. Vilei, "Evolution of wireless sensor networks towards the Internet of Things: A survey", in Software, Telecommunications and Computer Networks (SoftCOM), 2011, 19th International Conference on, vol., no., pp.1-6, 15-17 Sep. 2011.

[3] W. Mehmood, A. B. Mnaouer, A. Hassan, R. Tabish, B. Gaabab, F. Touati, "Performance Evaluation of 6LoWPAN Based Networks for Ubiquitous Health Monitoring System", in proceedings of the International Conference on Wireless Networks, July 21-24, 2014.

[4] D. Ugrenovic, G. Gardasevic, "Performance Analysis of IoT Wireless Sensor Networks for Healthcare Application", in proceedings of the 2th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN), 08–11 June 2015.

[5] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", IETF RFC 4919, 2007.

[6] T. Winter, P. Thubert, A. Brandt, T. Clausen, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, and J. P. Vasseur, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", IETF RFC 6550, 2012.

[7] R.T. Fielding, "Architectural styles and the design of network-based software architectures". Ph.D. dissertation, University of California, Irvine, California, 2000.

[8] Z. Shelby, K. Hartke and C. Bormann, "The Constrained Application Protocol (CoAP)", IETF RFC 7252, 2014.

[9] W.T Sung, K.Y. Chang, "Evidence-based multi-sensor information fusion for remote healthcare systems", Elsevier/Sensors and Actuators A: Physical, Company, Dec. 2013.

[10] F. Touati, R. Tabish, A. Ben Mnaouer, "Towards u-health: An indoor 6LoWPAN based platform for real-time healthcare monitoring", in Wireless and Mobile Networking Conference (WMNC), 2013, 6th Joint IFIP, vol., no., pp.1-4, 23-25 Apr. 2013.

[11] H. A. Khattak, M. Ruta, E. Di Sciascio, "CoAP-based Healthcare Sensors Network: a survey", in Applied Sciences and Technology (IBCAST), 2014, 11th International Bhurban Conference on, vol., no., pp.499-503, 14-18 Jan. 2014.

[12] M. Castro, A. J.Jara, A. F.Skarmeta, "Enabling end-to-end CoAP-based communications for the Web of Things", Elsevier/Journal of Network and Computer Applications, 2014.

[13] A. Dunkels, B. Grönvall and T. Voigt, "Contiki - a Lightweight and Flexible Operating System for Tiny Networked sensors", in Local Computer Networks, 2004, 29th Annual IEEE International Conference on, vol., no., pp.455-462, 16-18 Nov. 2004.

[14] N. Tsiftes, J. Eriksson, and A. Dunkels, "Low-Power Wireless IPv6 Routing with ContikiRPL", in proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, vol., no., pp.406-407, Apr. 2010.

[15] F. Österlind, A. Dunkels, J. Eriksson, N. Finne and T. Voigt, "Cross-level Sensor Network Simulation with Cooja", in Local Computer Networks, Proceedings 2006, 31st IEEE Conference on, vol., no., pp.641-648, 14-16 Nov. 2006.

[16] Tmote Sky datasheet, http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf

[17] A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol", Technical Report T2011:13, Swedish Institute of Computer Science, Dec. 2011.

[18] M. Kovatsch, S. Duquennoy, A. Dunkels, "A Low-Power CoAP for Contiki", in Mobile Adhoc and Sensor Systems (MASS), 2011, IEEE 8th International Conference on , vol., no., pp.855-860, 17-22 Oct. 2011.

[19] M. Kovatsch, "Demo abstract: Human-CoAP interaction with Copper", in Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011, International Conference on, vol., no., pp.1-2, 27-29 June 2011.