Communicating With Things – An Energy Consumption Analysis

Gérôme Bovet[1] and Jean Hennebert[1,2]

[1]Institute of Information and Communication Technologies
University of Applied Sciences of Western Switzerland
Boulevard de Pérolles 80, 1705 Fribourg, Switzerland
[2]Department of Informatics (DIUF), University of Fribourg
Boulevard de Pérolles 90, 1700 Fribourg, Switzerland
{gerome.bovet,jean.hennebert}@hefr.ch
http://www.eia-fr.ch

Abstract. In this work we report on the analysis, from an energy consumption point of view, of two communication methods in the Web-of-Things (WoT) framework. The use of WoT is seducing regarding the standardization of the access to *things*. It also allows leveraging on existing web application frameworks and speed up development. However, in some contexts such as smart buildings where the objective is to control the equipments to save energy, the underlying WoT framework including hardware, communication and APIs must itself be energy efficient. More specifically, the WoT proposes to use HTTP callbacks or WebSockets based on TCP for exchanging data. In this paper we introduce both methods and then analyze their power consumption in a test environment. We also discuss what future research can be conducted from our preliminary findings.

Keywords: Web-of-Things, Smart Buildings, RESTful services, Green Computing

1 Motivation

Citizens nowadays are becoming more and more sensitive to their environmental impact. Further to this, the energy prices are increasing over the years. Many countries have defined clear objectives for reducing the CO2 emissions and lowering the energy consumption not only for industries but also for households. A promising identified direction is to bring intelligence into buildings by optimizing how the energy is consumed.

We are currently developing a WoT based platform to be used in the context of smart-building applications. This platform is developed through an open-source project called Watt-ICT and is illustrated on Figure 1. The platform is composed of sensors, actuators, visual energy feedbacks and energy hubs that are all considered as things in the WoT approach. The energy hubs, called here GreenPC's, centralize the data and also act as smart gateways. Different kind

Pervasive 2012 Posters
Newcastle, UK, 18-22 June, 2012

of sensors are already integrated in our platform, able to measure, for example, temperature, presence and electric energy consumption. We have also integrated energy feedback lamps aiming at providing ambient information on the level of energy consumption. According to the size of the buildings, we envision that the GreenPCs may be distributed. Also, algorithms will be implemented in the GreenPCs to analyze and propose ad hoc actuation and control rules of the building equipments.

In order to allow an easy integration of our platform into an existing environment (house, building, office, etc.), we base our architecture on the one proposed by Guinard [1], considering things as parts of the Web. In this Web-of-Things (WoT) approach every component or *thing* is seen as a resource exposing REST services for communicating values (sensing) and manipulating its state (actuating). In such a framework, things may become more and more numerous with an increasing cost of the communication.

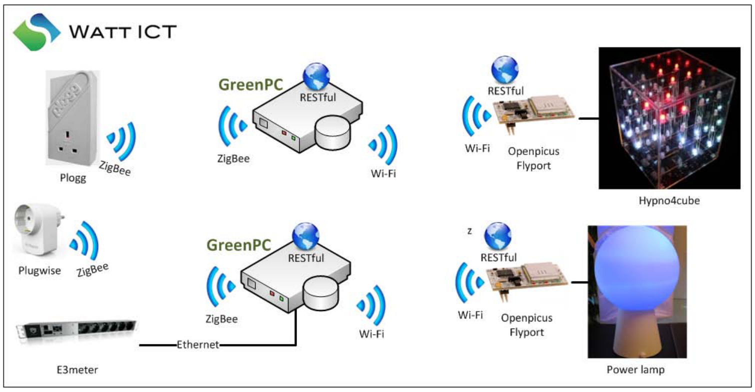

Fig. 1. Schema of the Watt-ICT platform

In the current WoT approach, two methodologies are proposed for managing event-based communications: HTTP callbacks and WebSockets. In the first case all exchanges are done by calling the corresponding REST service, providing the data in a HTTP packet. The second alternative is to keep a TCP connection open as long as possible between the producer and the consumer. Previous research [2][3] has showed the advantage of using WebSockets from a CPU point of view thanks to the fact that there is no HTTP overhead.

In our research we focus on the consequences of energy consumption of things. Things like sensors or actuators are often battery powered, and require an efficient energy management to make them available as long as possible. Our results will allow choosing the most suitable communication principle for energy aware things.

2 Implementation

We have implemented both event-based communication principles on an OpenPicus Flyport Wi-Fi module, that is used to connect things to the Web [4]. The small footprint Wi-Fi module integrates a TCP/IP stack and can embed a simple Web server. This module is 5V powered, based on a Microchip PIC24F controller and comes with an open source IDE and API for rewriting its firmware. An isolated test environment composed of a Wi-Fi access point, the OpenPicus Flyport module as the thing, and a computer acting as GreenPC was set up. To measure the power consumption of the Flyport module precisely, we used the Hameg HM815-2 Power Meter [5]. This power meter allows recording the measurements on a computer.

3 Results

Both communication methods were tested separately by sending a JSON payload of 46 bytes to the Flyport module at 500ms interval during 5 minutes. The average power consumption was measured over this period. From Figure 2 we can see that using a TCP connection is less power expensive than HTTP. The HTTP method consumes 5.3% more power and thus energy as only using TCP.

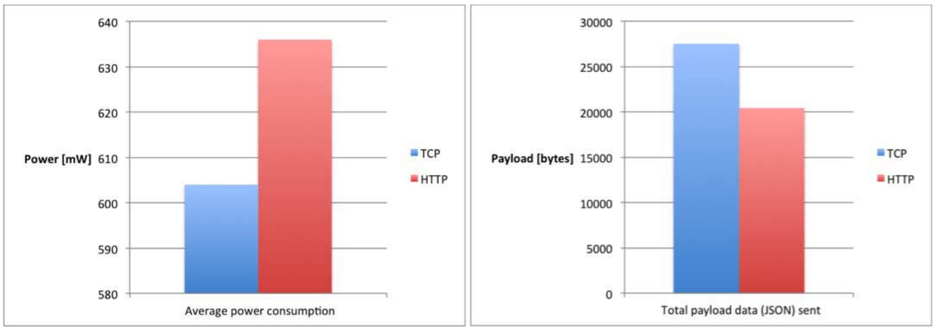

Fig. 2. The left part of the figure shows the average power consumed during the test for both communication protocols. The right part shows the volume of payload data (JSON) exchanged during the test.

The results can be explained by the number of packets exchanged in order to send the JSON payload. While in TCP mode only 2 packets are needed once the connection is established and maintained open, there are up to 18 packets in HTTP mode every time the JSON data is sent (as illustrated in Figure 3). In HTTP mode, as a consequence, the Flyport's radio transmitter is used over a longer period, thus increasing the power and energy consumption. In addition to this, less JSON data (-25.7%) could be exchanged with HTTP. The reason for it is due to the Flyport module having few computing power (16 MIPS) at disposal for handling the HTTP overhead.

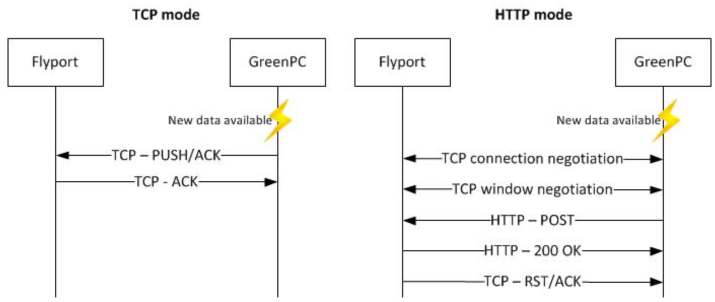

Fig. 3. Packets exchanged for the TCP and HTTP modes.

4 Future work

As previously illustrated, using the TCP method in our case allowed to save around 5% energy while allowing for a larger quantity of requests per unit of time. Surprisingly, the observed energy savings by using TCP instead of HTTP is not so large. In our future works, we will investigate further the reasons of such observations and analyze the impact of the payload and frequency of requests on the energy consumption. Knowing this function would allow to better choose how a thing should be powered (battery, solar panel, etc.) as a function of its usage.

References

[1] Guinard, D.: A Web of Things Application Architecture – Integrating the Real-World into the Web
[2] Priyantha, N., Kansal, A., Goraczko, M., Zhao, M.: Tiny Web Services : Design and Implementation of Interoperable and Evolvable Sensor Networks Proceedings of the 6th ACM conference on Embedded network sensor systems, 253–256
[3] Groba, C., Clarke, S.: Web services on embedded systems - a performance study 8th IEEE International Conference on Pervasive Computing and Communications Workshops PERCOM, 726–731
[4] OpenPicus Flyport Wi-Fi module: http://www.eikonsite.it/openpicus/prodotti.aspx?cprod=015350
[5] Hameg HM8115-2 power meter: http://www.hameg.com/0.147.0.html