Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 94 (2016) 137 – 143

Procedia Computer Science

The 13th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2016)

Management and Internet of Things

Mayra Samaniego, Ralph Deters*

mayra.samaniego@usask.ca, deters@cs.usask.ca

University of Saskatchewan, Dep. Computer Science, Saskatoon, Canada

Abstract

The study and development of Internet of Things (IoT) applications, web and mobile, is on the increase. Applications, working with data obtained from different areas such as transportation, smart homes, health care, public services, industry and many others. Previous studies have focused on managing the obtained data. However, managing the heterogeneous resources that get that data is an area that demands more attention. This work addresses the management of resources in the Internet of Things. This is achieved by proposing a virtual-resource edge layer, which enables access and configuration to constrained physical resources. The architecture presented focuses on the use of virtual resources as a management concept and identifies different approaches in the performance evaluation on edge computing devices. Using the IoT protocol CoAP, virtual resources are exposed in the edge network. An evaluation of a Go CoAP virtual resource is presented.

Peer-review under responsibility of the Conference Program Chairs
Keywords: Internet of Things; IoT Management; Virtual Resources; CoAP

1. Introduction

In the Internet of Things[1,2], the communication and resource management have been performed based on the technology in which the constrained network is implemented [6,7,8]. Each vendor has made sure of developing its own encapsulated-compatible and propietary communication standards. Applications have been supported by those propietary standars to get reliable information in an efficient and secure manner from the constrained environment to the cloud [3,4,5]. This has influenced users to adopt a certain technology in entire implementations to facilitate the duties of management (ITU-T M.3400). There has been some studies on facing IoT physical devices, but their main goal has been to facilitate the job of developers [10,11]. Nowadays, there is not a solution to allow access to diversified IoT network resources to get their information. This causes limitations in fulfilling the vision of the Internet of Things[9]. This work extends that vision not only having interconnected heterogenous ”things”, but also getting information in real time about those ”things” that are working in the environment. The architecture put forward is focused on abstracting the complexity of access and configuration of heterogeneous constrained networks resources to the concept of virtual

* Corresponding author. Tel.: +1-306-966-2072 ; fax: +1-306-966-4884.
E-mail address: deters@cs.usask.ca

1877-0509
Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.08.022

resources, offering a simplified management option. This work proposes the creation of virtual resources running in edge computing devices. Also, this study adopts the Constrained Application Protocol (CoAP) to communicate the virtual resources and the Go programming language to build them. This representation allows constrained components to be easily accessed from third parties. The direct communication with physical resources is broken up introducing virtual resource layers.

The remainder of the paper is structured as follows. Section 2 presents the definition of our IoT resources. Section 3 explains the experimental setup. Section 4 presents the performance evaluation of our prototype and the paper concludes with a summary in section 5.

2. IoT Virtual Resource Architectural Design

We define an IoT virtual resource as an abstraction of other resources, virtual or not. The architectural design of the IoT virtual resources is graphically illustrated in Fig. 1. Using the REST features of CoAP (RFC7252), a virtual resource exposes its own status and methods that interact with the states and can resolve interactions between its compound resources. The entire architecture comprises of three major layers namely: the view abstraction layer (VAL), the hardware abstraction layer (HAL) and the physical layer. The physical layer represents the sensors working in the environment. The two layers, hardware abstraction and view abstraction, are edge-hosted application middlewares, which are aimed at encapsulating the complexicity of access and configuration of physical resources. The HAL represents the lowest-level association between a physical resource and a virtual resource, resulting in a one-to-one relation, see Fig. 2. The virtual resources placed in this layer make use of CoAP verbs to receive orders to perform operations over their faced physical resources. The VAL acts as a bridge between final users and sensors. It represents an association between one or more HAL virtual resources and one or more VAL virtual resources, resulting in a many-to-many relation, see Fig. 3. The purpose of the virtual resources placed in this layer is to provide dynamic views to different kind of users with different access privileges. Also, this layer works as an edge-processing center in which the row-data is evaluated without going to the cloud. In this layer, virtual resources are divided into two groups, state-less and state-full virtual resources. State-less virtual resources, see Fig. 4, make use of the CoAP REST pattern to pull the states each time a request is received. State-full virtual resources, see Fig. 5, maintain the states of their compound resources in a database and make use of the CoAP observe pattern to receive updates of the resources they are observing.

This architecture is intended to cross the barrier of having only one level of communication, physical, when working with sensors.

3. Experimental Setup

This experiment was run over a VAL resource. The prototype involves a constrained network infrastructure, virtual resources and emulated clients. Once the VAL resource receives a CoAP request from the CoAP client, it gets the state of its compound resources and sends the processed information in a CoAP message. A Raspberry Pi with Go language and CoAP protocol was used as the edge computing layer, to run the virtual resource, see Fig. 6. We built the virtual resource and clients using the channels and routines features of Go, see Fig. 7. We set up the proposed architecture with devices with the following specifications: Raspberry PI 2 Model B with. processor ARMv7, CPU: ARM Cortex-A7 quad core - 900 MHz, RAM: 1GB, Operating system: Raspbian OS. IMac with processor: Intel Core i7 - 3.5 GHz. RAM: 16 GB, 64 bits operating system. The clients connected to the virtual resource over a dedicated WiFi connection.

4. Evaluation

To evaluate the usefulness of our virtual resource we measured its throughput and response times. We evaluated the behavior of our virtual resource testing two processes, the Core Link Format (RFC6690) discovery of services through a well-know interface and the current state retrieving. These evaluations were performed from the client perspective. Also, we evaluated the database connection time in the VAL state-full resource. The services that our virtual resource