IoT Device Management and Configuration

A Thesis Submitted to the College of

Graduate and Postdoctoral Studies

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In the Department of Computer Science

University of Saskatchewan

Saskatoon

By

YUNXIAO WANG

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other uses of materials in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

ABSTRACT

As the number of IoT devices grows, the management and configuration of IoT devices becomes crucial in resource constraint networks. It is hard to manage and configure a large amount of heterogeneous resource constraint IoT devices because people need to know how they connect to each other, what internet-enabled services are available to provide, and how people interact with things through the internet.

The thing-centric approach focuses on user experience when engaging things, but the cloud-centric approach switch the focus to IoT services that can process data streams collected from things and applications that help get people joined in the IoT world. To manage IoT populations effectively in a centralized manner, not only does it mean that moving computational power closer to the edge is a way to reduce bandwidth and latency, but it also implies that it is necessary to build an architecture which can scale and manage tons of connected devices by a uniform interface. In particular, RESTful Web services can provide a uniform interface that operates resources by HTTP methods. For example, users can read and write data by a uniform interface, and a flowerpot can write data and be triggered to water plants by a uniform interface. Thus, in the scope of IoT, embedded middleware can implement uniform interface by REST model.

Virtualizing physical things has emerged as a design pattern to build IoT systems. Resource less constraint devices are capable of being virtualized with enough CPU power, memory, networking, but they are more expensive and power consuming. However, resource highly constraint devices take advantage of low energy consumption and cheaper price, but they cannot be virtualized because they do not have ability to even run a single multi-threaded program. Therefore, it is very important to select the right platforms for the right roles. In our case, we use Raspberry Pi 3 as a middleware and Nordic nRF52832 as a BLE endpoint.

In this thesis, a REST-based IoT management system based on Service-Oriented Architecture is built, and the performance of the system has been tested, including the response time of HTTP GET and POST requests of the centralized server in a Fog domain and a script engine onto a BLE-enabled endpoint.

ACKNOWLEDGEMENTS

Hereby, I would like to thank for my family to support my study towards the Master of Science degree in the University of Saskatchewan.

Also, I would like to thank for my supervisor Professor Ralph Deters. Under his supervision, not only did I have a chance of academic review, but I also learnt lots of cutting-edge concepts in the field of Cyber Network and extended practical skills in programming. He is more like a life advisor than a knowledge giver.

Last but not least, I would like to appreciate any help from my MADMUC lab members who shared their experience and the view of problem solving. With the help of the above, I enjoyed my study through this unforgettable period in my life.

CONTENTS

PERMISSION TO USE i

ABSTRACT ii

ACKNOWLEDGEMENTS iii

CONTENTS iv

LIST OF TABLES vi

LIST OF FIGURES vii

LIST OF ABBREVIATIONS x

INTRODUCTION 1

1.1 Problem Definition 2

1.1.1 How to provide uniform web-like interface? 3

1.1.2 How can we use this interface to send commands to things? 3

1.1.3 How can we change code to run new functionality? 4

1.1.4 Research Goal 4

LITERATURE REVIEW 5

2.1 IoT 5

2.2 Hardware Platforms 7

2.2.1 The Concept of SoC 7

2.2.2 Introduction of SoCs 9

2.2.3 Summary 17

2.3 IoT Model 18

2.3.1 IoT Fog 18

2.3.2 SDN 19

2.3.3 Virtualization of IoT Fog 19

2.3.4 Restful Model 20

2.3.5 CREST 21

2.3.6 Summary 21

2.4 Protocols 22

2.4.1 Transport Layer Protocols 22

2.4.2 Application Layer Protocols 22

2.4.3 BLE communication layer protocol 25

2.4.3.1 BLE protocol stack 25

2.4.3.2 GATT 27

2.4.4 Summary 29

2.5 Solutions to Problems 30

ARCHITECTURE 32

3.1 Proposed System Architecture 33

3.1.1 Work Flow 35

3.1.2 RESTful Web Services 36

3.1.3 Script Engine 38

3.2 Summary 39

IMPLEMENTATION 40

4.1 RESTful Web services in Embedded Middleware 40

4.2 JavaScript Execution in BLE endpoints 43
4.3 Summary 45

EXPERIMENT **46**
5.1 Performance of Middleware 47
5.1.1 Performance of GET 47
5.1.2 Performance of POST 51
5.2 Performance of Script Engine 57
5.3 Summary 60

CONCLUSION **62**

FUTURE WORK **63**
7.1 Decentralization with Access Control 63
7.2 NFC (Near Field Communication) 64

REFERENCES **65**