

# A Data Collecting and Caching Mechanism for Gateway Middleware in the Web of Things

Xuchao Chang, Chunhong Zhang, Li Sun

Beijing University of Posts and Telecommunications, Beijing, 100876, China

E-mail: sklxc@gmail.com

**Abstract**—The Web of Things (WoT) is a refinement of the Internet of Things by integrating heterogeneous devices not only into the Internet (the network), but into the Web (the application layer). With the expansion of applications and devices, WoT Gateway Middleware is bearing more and more pressure from data processing, and therefore the WoT gateway middleware takes more time to respond to the requests of applications. A data collecting and caching mechanism is an effective solution to this problem. In this paper, we begin by describing the gateway middleware architecture with the focus on data collecting and caching functionality. Then the data collecting and caching (DCC) functionality module is discussed in detail. The DCC module supports three kinds of data collecting and caching modes: fundamental mode, configurable mode and adaptive mode. We finally demonstrate how these three modes work. A SMART HOME system is developed as the implementation to verify and discuss the proposed mechanism.

**Keywords**—Web of Things; WoT gateway middleware; data collecting and caching.

## I. INTRODUCTION

The Web of Things has been proposed to aim at fully integrating everyday devices and objects to the Web. Some Practical systems have also been developed. For example, SmartBUPT is a WoT platform in BUPT (Beijing University of Posts and Telecommunications) campus. It aims to create an open campus innovation platform to facilitate users to create useful and intelligent services in their daily campus lives by integrating and opening sensor resources via Web Service technologies [1].

The WoT gateway is the core device in WoT that will bridge the non-internet access physical devices (such as sensors, actuators, RFID tagged objects, etc.) to the Web and will abstract data and capabilities of these devices into programmable Web Service APIs [1]. WoT Gateway Middleware (GM) is used to implement the main features of the gateway, which simplifies the implementation of the applications, in order to abstract from the peculiarities of the operating system (OS) and hardware components and to provide services to manage and configure the WoT system [2][3].

Since the WoT GM can directly provide atomic Web Services of devices to end-users and other service consumers, it is bearing more and more pressure with the expansion of applications and devices. As a result, it takes more time for WoT GM to respond to the requests of applications. A data

collecting and caching mechanism is an effective solution to this problem. As an essential and fundamental feature of WoT GM, Data Collecting and Caching (DCC) is an important technology to reduce the noticeable response time perceived by users and ensure the experience of the WoT application users. Thought some work has been done for supporting data collecting and caching in Web and WSN, a large research effort is needed in the filed of data collecting and caching mechanism in the WoT.

The remainder of this paper is organized as follows: section II presents the challenges to guarantee real-time requirements in WoT. In section III, we introduce the gateway middleware architecture with the focus on data collecting and caching functionality. Section IV presents the architecture of Data Collecting and Caching (DCC) module and introduces the three kinds of DCC modes: fundamental mode, configurable mode and adaptive mode. In section V, we demonstrate how these three modes work. Then the implementation of our mechanism in a SMART HOME system is introduced in Section VI. Finally, conclusion and future work are given in section VII.

## II. CHALLENGES

Since WoT operate in the real world and in people's everyday life, real-time communication is a basic requirement for WoT systems. For example, an accelerometer might have to be read every 10 ms, else there will be a bad estimate of speed and consequently a probability of a vehicle crash [4]. Another typical example that requires real-time communication is in biomedical circumstances. It is extremely important that life-critical medical data is processed within a very short time, else tragedies might happen.

It remains a big challenge to guarantee the real-time requirement in WoT due to the following reasons:

### A. Large Scale

WoT is a huge global information system composed of hundreds of millions of devices and a wide variety of applications. With development of WoT technologies, the most important WoT application areas cover infrastructure construction, public security, environment protection, modern agriculture, intelligent industry, urban management, business service and other fields [5]. As a result, WoT GM is bearing more and more pressure and it takes more time for WoT GM to respond to the requests of applications.

### B. Resource Constraint

In contrast to traditional web resources, which are hosted on servers with strong capabilities and functionalities, the resources in WoT are hosted on resource constraint gateways. WoT gateways are generally resource constrained with respect to computation, storage, bandwidth, and power supply. Consequently, it becomes a great challenge to provide real-time service to applications.

### C. Devices Heterogeneity

WoT generally comprises a large number of different devices as well as distinct services, which may generate dramatically diverse data sources. Different data sources may have different real-time requirements. For instance, in SMART HOME system, when smoke sensor alarm, the alarm data should be processed as quick as possible, however, the data of wind sensor can be delivered and handled in a short while.

### D. Applications Heterogeneity

WoT supports various kinds of applications, which has different real-time requirements. As a result, WoT GM should dynamically modify its real-time policies when operate in different applications.

In order to deal with such problems, we propose a Data Collecting and Caching mechanism for WoT GM. We also implement the mechanism in a SMART HOME system for demonstration. It is detailed in the following sections.

## III. ARCHITECTURE OF WOT GATEWAY MIDDLEWARE

The WoT GM can help bridge the gap between the high level requirements from pervasive computing applications and the underlying operation of WoT, besides providing advanced functionalities for the management and configuration of the WoT.

According to [6][7], we establish our WoT Gateway Middleware architecture, as is shown in Fig.1. From bottom to top, the architecture includes three layers: the device abstraction layer, the control layer and the presentation layer.

### A. Device Abstraction Layer

The main function is to abstract from the low-level details and peculiarities of each individual device and provide unified access to sensors and actuators independent of their individual transfer protocols. Device Abstraction Layer is composed of device drivers.

### B. Control Layer

This layer embodies main functionalities and forms the core of WoT Gateway Middleware. The functional modules include: Data Collecting & Caching Module, Data Processing & Storage Module, Resource Management Module, Log Management Module, Device/Gateway Configuration Management Module, Event Management Module, QoS Management Module, etc. We will discuss the Data Collecting and Caching Module in detail in the next sections.

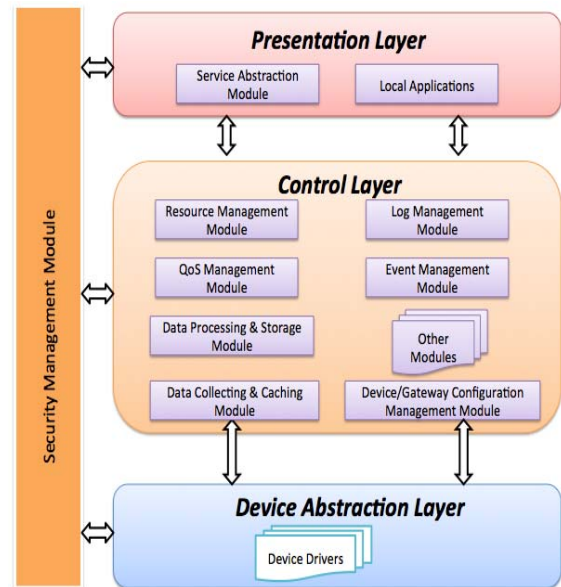


Figure 1. Architecture Overview

### C. Presentation Layer

Main function of this layer is making the WoT GM components accessible to the outside world. It is a thin layer on top of the Control layer managing requests from clients through a REST interface. This layer consists of two modules: Service Abstraction Module and Local Applications.

## IV. ARCHITECTURE OF DATA COLLECTING AND CACHING MODULE

As indicated in section III, Data Collecting and Caching (DCC) is an important functional module in WoT GM. DCC is responsible for collecting data required by the application and caching data to achieve high availability and faster query execution.

The Logical structure of DCC module is shown in Figure 2. The DCC module supports three kinds of data collecting and caching modes: fundamental mode, configurable mode and adaptive mode. Each mode has its own Data Collecting and Caching Policy (DCCP) file. The DCCP file is a formatted document, such as XML document, JASON document and customized formatted document, etc. The DCCP file of each mode is defined as follows: FundaDCCP for fundamental mode, ConfigDCCP for configurable mode and AdaptDCCP for adaptive mode. These three modes will be discussed in the next section. The Data Collecting and Caching module consists of two main layers: Policy Rule Function layer and Policy Enforcement Function layer. In the following subsections, each key component of the architecture will be detailed.

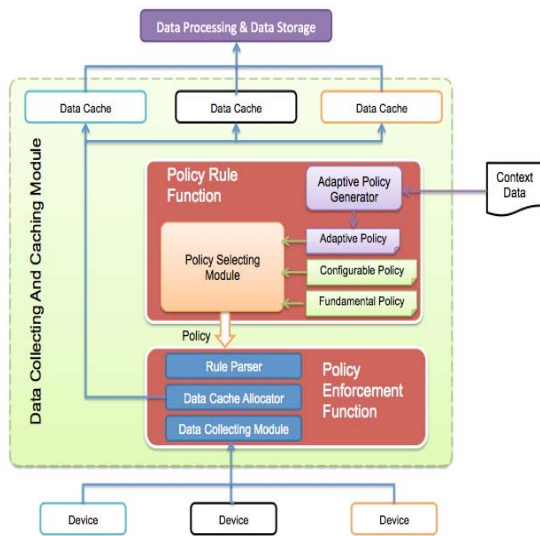


Figure 2. Architecture of Data Collecting and Caching module

### A. The Policy Rule Function Layer

The main function of this layer is to generate or modify three kinds of policies and determine which kind of policy to be adopted.

**Adaptive Policy Generator (APG):** generates AdaptDCCP based on Context Data in WoT Gateway Middleware. The Context Data includes: environmental parameters (such as weather, time, season etc.), device profile and application profile.

**Adaptive Policy:** is the policy file used in adaptive mode, namely AdaptDCCP. It is generated by Adaptive Policy Generator.

**Configurable Policy:** is the policy file used in configurable mode, namely ConfigDCCP. It can be configured and modified in two ways. First, GM administrator can modify ConfigDCCP through configuration page. Second, application users can modify ConfigDCCP via Policy Configuration Service (PCS) provided by GM.

**Fundamental Policy:** is the policy file used in fundamental mode, namely FundaDCCP. It provides basic real-time configurations for GM.

**Policy Selecting Module (PSM):** provides policy selection. As mentioned earlier, there are three data collecting and caching mode, and each mode has its own DCCP file. Select a mode is to select the corresponding DCCP file.

### B. The Policy Enforcement Function Layer

Main function of this layer is parsing the adopted policy to data collecting and caching rules and then executing the rules.

**Rule Parser (RP):** parses the adopted policy to data collecting and caching rules (DCCR). DCCR consists of two parts: allocation of what size of data cache, which method to be adopted to collect data. GM should allocate

suitable data cache size for different devices mainly based on the following reasons. First, save the gateway resources. Second, different devices generate data in different frequencies and data sizes. The method to collect data from diverse devices varies. For event-based devices, GM should use Pub/Sub paradigm to implement data collecting. For query-based devices, GM may use poll to collect data. The frequencies to collect data are also different for different devices and applications.

**Data Cache Allocator (DCA)** decides what size of data cache to be allocated to each device based on DCCR.

**Data Collecting Module (DCM)** decides which collecting method to be adopted and in what frequency to collect data from each device based on DCCR.

Figure 3 shows the proposed interaction sequence of data collecting and caching. Assuming FundaDCCP, ConfigDCCP and AdaptDCCP have been generated, first, PSM select a policy and pass it to RP where the policy is parsed to rules. Then the rules are passed to DCA and DCM. DCA allocates specified size of data cache to each device based on rules. Finally, DCM collects data from each device through the designated method and put data into data cache.

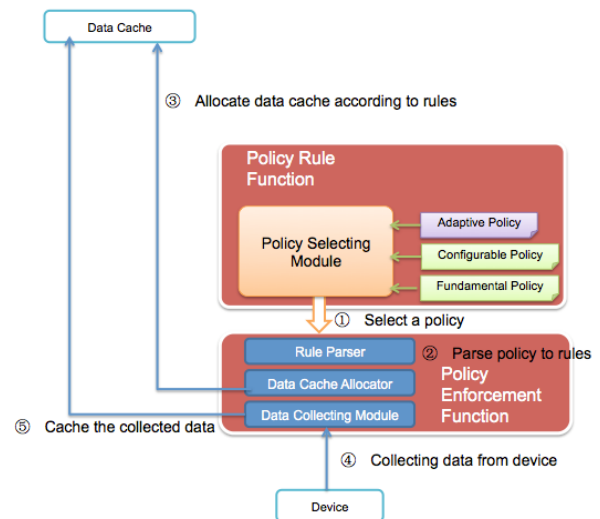


Figure 3. Interaction sequence of data collecting and caching

## V. THREE MODES OF DATA COLLECTING AND CACHING

As mentioned in section IV, there are three kinds of data collecting and caching modes: fundamental mode, configurable mode and adaptive mode. In this section, we will discuss the details of these modes.

### A. Fundamental Mode

Fundamental mode is the default mode used by GM to collect and cache data. After restarting GM, it will use fundamental mode. Fundamental mode provides basic real-time configurations for GM.

### B. Configurable Mode

When different applications use the same set of gateway and devices, the applications probably have different real-time requirements. Therefore, the data collecting and cache policy must be able to change according to the demands of the application. Configurable mode provides an efficient solution to this problem. The ConfigDCCP is initialized to FundaDCCP. When FundaDCCP cannot meet the real-time requirement of a certain application, there are two ways to change ConfigDCCP. First, GM administrator can modify ConfigDCCP through configuration page. Second, application users can modify ConfigDCCP via Policy Configuration Service (PCS) provided by GM. The interaction sequence of the second method is shown in Fig 4.

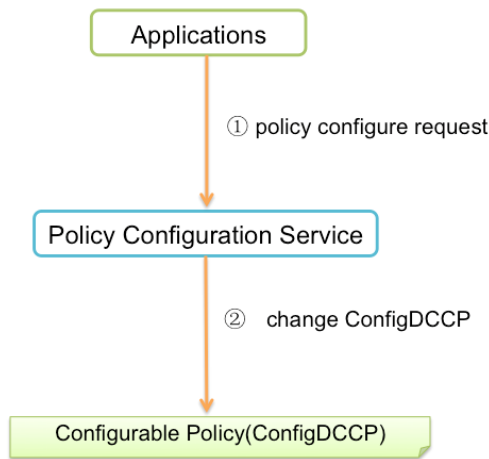


Figure 4. Interaction sequence of changing ConfigDCCP via PCS

### C. Adaptive Mode

To autonomously adapt to different real-time requirements of applications, the GM shall have self-configuration or advanced cognition capability. In this sense, the HGW is able to intelligently change its AdaptDCCP based on Context Data. The Context Data includes: environmental parameters (such as weather, time, season etc.), device profile and application profile. As the same as ConfigDCCP, The AdaptDCCP is initialized to FundaDCCP.

## VI. IMPLEMENTATION

We have proved our architecture in a SMART HOME system, which was based on an aging care cooperation project between China and Finland. In this system, several kinds of devices are deployed in a demonstration room. Fig 5(a) shows the layout of the demonstration room and device deployment diagram. The devices include: temperature sensors, humidity sensors, light sensors, wind sensor, Smoke alarm, Intrusion detector, light control relay, etc. All these devices are integrated to Web by a WoT gateway. As shown in Fig 5(b), WoT gateway is an industrial PC, which has a Windows operation system. It can connect the devices

mentioned above via Ethernet, WiFi, serial ports, Zigbee and RF433 through communication modules deployed on it. The GM is deployed on the WoT gateway to integrate all these devices into the Web and it uses .NET Framework as the runtime environment. Besides from integrating devices to web, GM also provides advanced functionalities for the management and configuration of the WoT system. Configurations are implemented by a configuration web page (Fig 5(c)) which is a part of GM. The configuration of DCC policies is also implemented through this page. As proof of concept, we develop multiple applications based on our architecture and mechanism. The applications include both web applications (Figure 5(d)) and smart terminal (e.g. iPad, android devices, etc.) application (Figure 5(e)). Our GM support all three data collecting and caching modes.

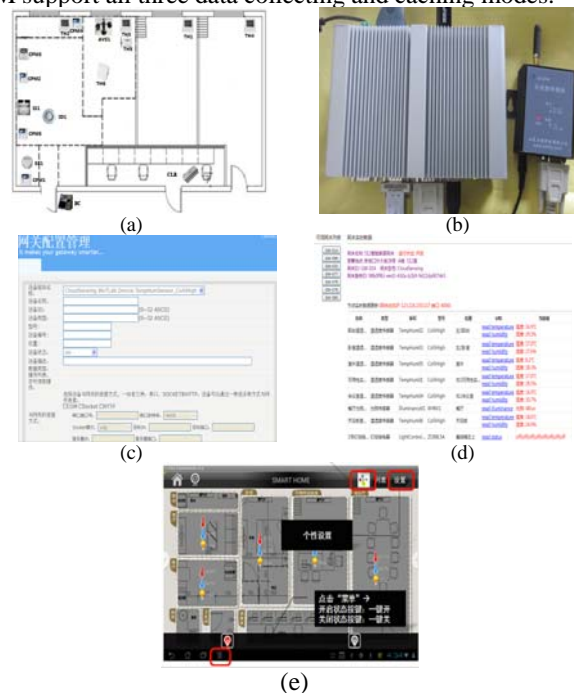


Figure 5. Overview of our demonstration. (a) system layout; (b) gateway; (c) GM configuration page; (d) web application; (e) smart terminal application

## VII. CONCLUSION AND FUTURE WORK

In this paper, we have introduced a data collecting and caching mechanism for the WoT gateway middleware. The main objective of this the proposed mechanism is to meet the real-time requirements of WoT. With the implementation of SMART HOME system, we have proved the feasibility and reasonability of the mechanism. Therefore, it can be widely used in smart home, industrial monitoring, smart grid, environment monitoring etc.

In future work, we will concentrate more on the other aspects of QoS management including fault-tolerance, reliability and security.

#### ACKNOWLEDGMENT

This work was supported by the following projects:

- (1)Project on the Architecture, Key Technology Research and Demonstration of Web-based Wireless Ubiquitous Business Environment (No. 2012ZX03005008-001);
- (2)China-Finland Cooperation Project on the Development and Demonstration of Intelligent Design Platform Driven by Living Lab Methodology(No. 2010DFA12780);
- (3)Collaborative Network Control Platform and the Key Technology of Ubiquitous Network Under Multi-terminal(No. 2011ZX03005-004-04).

#### REFERENCES

- [1] Zhenyu Wu, Timo Itala. Gateway as a Service: A Cloud Computing Framework for Web of Things. 19th International Conference on Telecommunications (ICT 2012)
- [2] Wang MM, Cao JN, Li J et al. Middleware for wireless sensor networks: A survey. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 23(3): 305–326 May 2008
- [3] L. Atzori, A. Iera and G. Morabito, "The Internet of Things: A Survey", *Computer Networks*, Vol. 54, No. 15, pp. 2787-2805, Oct. 2010
- [4] John A.Stankovic, Tarek Abdelzaher, Chenyang Lu, Lui Sha, Jennifer Hou. Real-Time Communication and Coordination in Embedded Sensor Networks.
- [5] Qian Zhu, Ruicong Wang, Qi Chen, Yan Liu and Weijun Qin. IOT Gateway: Bridging Wireless Sensor Networks into Internet of Things. 2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing
- [6] Vlad Trifa, Samuel Wieland, Dominique Guinard, Thomas Bohnert, "Design and Implementation of a Gateway for Web-based Interaction and Management of Embedded Devices"
- [7] Hao Chen, Xueqin Jia, Heng Li. A Brief Introduction to IOT Gateway. Proceedings of ICCTA2011