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.