

# Abstract

The Internet of Things(IoT) edge devices has limited resources, such as limited storage, RAM, CPU performance, and bandwidth. However, the software that runs on them may occasionally need to be updated, and this kind of updating is a challenge. Since IoT devices are resource-constrained, this dissertation would like to explore an approach that reduces the update package size, improves the update efficiency, and considers this approach's robustness.

There are already many business solutions and research on this topic, but they mainly focus on monolithic updates, i.e., overwriting the previous firmware entirely. However, improving the current monolithic solutions is challenging to gain a better result because of many physical level constraints—for example, chips' frequency, Bluetooth bandwidth, and power supplies. Therefore, instead of working on current monolithic update solutions to approach the hardware performance limitation, we would like to find a different way to apply updates.

In this dissertation, we design three components to implement an incremental updating system. The first component is a reversed FAT file system called rFAT. This file system merges multiple device flash partitions into one, allowing one firmware to occupy more spaces and naturally support firmware downgrade. The next is a different algorithm, BSDiff-Inplace, to apply an incremental update. A different algorithm can generate a patch file, and the device can recover the new firmware by using this patch and the old firmware. Moreover, BSDiff-Inplace is explicitly designed for tiny devices, requiring less RAM than other difference algorithms like BSDiff, EXEDiff, and Courgette. Last, because flash chips only support page-wise read/write, we implement an EEPROM emulator to support random read/write.

Overall, this project accomplishes the main goals. For instance, users can generate a mini update patch, send it to a device, and apply it. However, the performance of BSDiff-Inplace can still be better, as its compression rate is only 50% of the compression rate of the original BSDiff. Besides, in the future, exploring modular updates or a hybrid of incremental and modular updates can improve the performance.