Using Wake-on-LAN to Manage Devices

Thomas Smith
tcs1g20@soton.ac.uk

1 Introduction

This project will look at how the Wake-on-LAN (WoL) standard can be implemented into home networks to improve accessibility and reduce power consumption of devices.

Some of the example use-cases of this project include:

- managing a home network of devices that consists of personal and IoT devices,
- accessing devices in hard to reach locations, and
- scheduling devices to wake and perform specific actions.

By only having devices on when they are in-use, this project aims to massively reduce the amount of power wasted everyday by idle devices.

2 Wake-on-LAN

Wake-on-LAN is a network standard that allows remote devices to be turned on by sending a *magic packet* to it. It was first introduced in 1998 [1] by the Advanced Manageability Alliance AMA; this alliance consists of several large technology companies that was created with the purpose of creating standards that helped streamline computer management. Some examples of standards introduced by the AMA are: Desktop Management Interface [2], Alert Standard Format [3], and Common Information Model [4].

2.1 Magic Packet

A magic packet is a frame with a 102 byte payload that consists of: 6 bytes that are all *0xff*, and 16 copies of the target device's MAC address. The target device will be listening specifically for this packet whilst in a low-power mode; when it is broadcast over the network the target device's network interface card will send a command to the power supply or motherboard to wake the system up.

2.2 Limitations

The WoL standard is limited by:

- not having any form of delivery confirmation so there is no way to tell if the device was actually woken without interfacing with it directly,
- requiring knowledge of the device's MAC address to be able to wake it, and
- being limited to IP-based devices.

3 Design

3.1 API

This application will include a HTTP endpoint that will allow users to access it remotely over the internet.

3.2 Events

This section will detail the various events offered by this application that can be used to wake specific devices.

3.2.1 Schedules

Users will want to schedule devices to turn on at recurring periods. This application will offer users the ability to set recurring days of the week and times at which a device should be woken.

3.2.2 Scanning the LAN

The vast majority of users will have a personal device, such as mobile phone, that will connect to their home router when they're nearby. This means that by listening for when devices join the local network, this application can determine when a user arrives home and can wake the relevant set of devices.

Checking whether a device is in the network

can be achieved using an ARP request and assuming that if the device doesn't respond then it can't be in the network.

The user will configure devices to wake when they're personal device joins the network. Some examples would be their PC or various IoT devices around the home.

3.2.3 Bluetooth

Bluetooth is a short-range, low-bandwidth, low-power wireless communication protocol that is commonly found in battery powered devices. By allowing this application to search for bluetooth devices, it massvely increases the number of devices we can search for and removes the requirement from the previous section of being in the same LAN.

To initiate a connection, a device will send an *inquiry* message to the target and if the target is in discoverable mode then it will respond. A non-discoverable device will never respond to these messages; however, the paper *Detecting Non-Discorable Bluetooth Devices* [] showed that non-discoverable devices can be detected using an 'enhanced brute force search attack' as long as you know the device's address. But, this solution is time consuming and wouldn't be a useful addition to this proejct.

3.2.4 Environment Sensors

It may be useful for devices to be triggered by environmental factors, such as detecting motion, room temperature, and light.

4 Implementation

4.1 API

The API was written using FastAPI []. One main advantage to this was that FastAPI generates and hosts OpenAPI documentation; this means users can easily explore and interact with the API.

The API will be secured using OAuth2 and JWT tokens, where users will have to enter a username and password before being able to view any data or make any changes.

4.2 Data Storage

This application will create a local SQLite database consisting of the following tables:

- **devices** Information about the devices that helps us identify them. This includes their MAC address, static IP address, a user-provided alias, and whether or not they can be woken with a magic packet.
- **devices_to_search_for** This will store information about devices that we are searching for through various protocols. Each record references a device from the **devices** table and whether or not they should be searched for through various mediums (such as Bluetooth or in the LAN).
- **schedules** This will detail the recurring days of the week that a device should be woken up.
- **services** This includes details about all the background processes that will be run. For example, whether the application should search for Bluetooth devices.
- **users** Stores information about the users of the system, where a single user will be an admin.

4.3 Other Details

4.3.1 Environment Sensing

For this project, the Enviro sensor for Raspberry Pi [] was used. This allowed me to trigger events when the following metrics were above and below given thresholds:

- the current room temperature,
- the light level,
- nearby noise, and
- motion.

References

[1] *IBM News room - 1998-04-15 IBM Announces Universal Management – Industry's Most Comprehensive Tools to Lower Total Cost of Ownership - United States.* Oct. 12, 2012. URL: `https://web.archive.org/web/20121012155338/http://www-03.ibm.com/press/us/en/pressrelease/2705.wss` (visited on 05/07/2023).

[2] *DMI — DMTF*. URL: https://www.dmtf.org/standards/dmi (visited on 05/07/2023).

[3] *ASF — DMTF*. URL: https://www.dmtf.org/standards/asf (visited on 05/07/2023).

[4] *CIM — DMTF*. URL: https://www.dmtf.org/standards/cim (visited on 05/07/2023).