Real Time Operating Systems Project Report

Real Time Location Tracking Using WiFi inside an Organization

Sravya Goli
IMT2014020

Vineet Reddy Rajula
IMT2014045

April 29, 2018

1 About the project

Our project is like the "Marauder's map" in Harry Potter. Google maps can help us locate a person on road, it can give us the precise location with the building name but it can't help us locate the person inside a building or inside an organisation. Our project can help us do this. Locating people inside a building or an organisation. In other words, this project can be used on top of google maps, to locate inside a building.

2 Implementation

Nowadays people carry their cell phone everywhere. We used this as an advantage in building our project. We created an android application which on click of button enables authorised people to locate the person.

Every organisation or a building has wifi access points. Almost every time these access points are set up in such a way that at any point in the organisation or building is wifi-accessible. This is done by installing multiple access points in the organisations. Therefore, when people move inside the building or an organisation, the wifi access points keep changing. We are using the changing of access points to locate people. Our application can determine the radius in which someone is located.

3 Initial Thoughts

Initially, we thought of simulating a wireless environment using a bunch of Raspberry Pi's and use a wireless connection enabled device to move around. We wanted to do this, so that we can have control of the bssid of the access points, but upon some study we got to know that it is not difficult to obtain the details of a access point a device is connected to. Basic details like bssid, MAC address, IP etc of an access point can be obtained easily.

So, we decided to start the project using the existing wifi access points on the campus. The plan was to make an android application which gets the details of the connected access point and sends the details to a server which will use the data to display the location on a map. As the data needs to be sent regularly, the android app has to be run in the background.

4 Technical Details

4.1 Map

We created a simple map of the ground-floor of the old academic block. Then we noted down the locations of the access points and their mac addresses, bssids present on the floor. We located 3 access points in total on the floor as shown in the figures below 1a 1b.

4.2 Database

To store the current location and previous location, we created a mysql database. The database table has only two entries, a UID and the BSSID. The android app does a GET request to a php script running on the server, which will then add the data to the database.

4.3 Dashboard

Once we had the database and the map, we used the image of the map and rendered it using HTML and CSS. To change the access point on the map, we wrote a javascript function. The whole page is then served using a NodeJS server.

4.4 Android

We have created a simple UI with a button as shown in 2. Clicking this button starts a service that runs in the background. The background service runs infinitely checking for a change in the connected wifi access point. If change is detected then it sends changed access point's mac id and user id to mysql database.

4.5 AWS

As the wifi on the app keeps changing, we couldn't host the server locally. Though we tried forwarding the packets to a public ip using 'ngrok', we needed two ports while ngrok free version supports only one. So, we decided to host everything on aws. We applied for the aws credits with the help of Github

Student Developer pack.

So, we hosted the NodeJS server & MySQL server on an aws EC2 instance. We then used the public IP to send data on android.

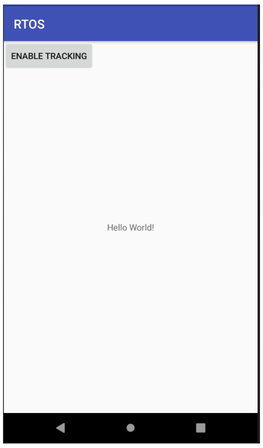

Figure 1: Andorid app UI

5 Working

As a whole, whenever the mobile connects to a new access point, the app running in the background of the same mobile sends the data as a GET request to a PHP script running on a apache server on AWS. The script then adds the data to a MySQL server, which will then be displayed on the map using HTML, Javascript and CSS which are running on a NodeJS server. The same repeats.

6 Real Time

There are two real time concepts in this project. One is the android app which is running as a thread and sending the data only when there is a change in the wifi bssid. The other is displaying the location on the map without much delay. For this there were two options, one is to make the Javascript (client) ask the NodeJS server for new changes in database. This is polling, and needs to happen repeatedly after a certain time interval. Or whenever there is a change in the database, the server has to send a notification to the client, to update the location. This is a pushing. We went with the second method as it seemed more efficient.

So we used websocket library called 'socket.io' and a 'mysql-event listener' library in the NodeJS server. The socket.io is used to connect server and the client through websockets and this very similar to socket programming. The mysql event listener can detect when a new data record has been added to a specified database and table. When the event happens, the websocket library socket.io sends a notification to the client. It is then received on the client side using javascript which then updates the location accordingly.

7 Limitations And Future Work

Though technically there isn't any delay in updating the location, when moving from one side to another at a constant pace there is a bit delay in the mobile trying to connect to the nearest network i.e delay when mobile has to change from one network to a another network. Due to this there is a small delay in updating the location. In fact it varies from mobile to mobile.

We have done this for a single floor. This can be extended to multiple floors. One good use case for this would be to identify a person in distress in a building. usually it is difficult to track a person inside a building, using this the person in distress can press a distree button on his phone, which can alert the appropriate authorities and the person can be helped.

8 References

1. Real Time Engine socket-io

2. A Node JS NPM package that watches a MySQL database and runs callbacks on matched events. mysql-events