

Challenge 7: Localization

1 Challenge

Design and implement a localization algorithm in java or c++ which shows the location of your laptop based on the received signal strength of wifi access points. Your program will be provided with a list of (MAC address - rssi) pairs of all access points your wifi adapter can see. You will use this data combined with a list of (MAC address - location) pairs we provide to find the location of your laptop.

2 Details

You are given a map of all WLAN access points in OH 115 and 116. Using the location and MAC address of these access points you need to design and implement a localization algorithm which determines where your own laptop is.

Consider the following issues while designing your solution:

- Think about how you can calculate the distance between your device and the access points. Correct modelling of the signal strength to distance is not trivial and highly depends on the environment and the device, antennas, etc.
- Triangulation is a common approach for finding positions. For doing this you need to use at least three access points. The more access points you use, the more accurate your localization will be.
- There are various ways to increase the accuracy of your localization algorithm even further. You may think of fingerprinting of signals of all access points and mapping the location of your device to them.
- There is a trivial way to find a location that is at least somewhat accurate. You might want to try implementing a simple solution first!

We provide you with a framework that measures the strength of all reachable access points once per second. When new measurement data is available a method in your implementation will be called to calculate a position. The framework reports your calculated position to the netsys webserver (<http://netsys.ewi.utwente.nl/challenge7/>) where it will be drawn on a map. The java version also draws the location on a local map.

- To use the provided code you will first need to start the “WlanServer.exe” application (Windows) or run the wlanServer java application (Linux). Further instruction for starting these applications can be found in the readme file with the software. The provided framework will connect to the WlanServer and receive the signal strength data. Please note that due to limitations in the wireless drivers the Linux scanner might not update as often as the Windows version!
- Java: Your assignment is to implement the *LocationFinder* interface (in the Location package). C++: Your assignment is to implement a *LocationFinder* in the protocol directory.
- Java: Change the *locationHandler* in *WlanScanner.java* to your own implementation. C++: Change the *SimpleLocationFinder* class to your own class in *main.cpp* (keep the variable name the same!).
- The provided code will call the *locate* method in your *LocationFinder* each second (so don’t put any unending loops in your code!).
- *locate* should return a *Position* object (from the Utils package/directory). This Position will be printed to your console and transmitted to the map.
- If the *send* boolean is true, the location will also be sent to our website (not optional for c++ version): <http://netsys.ewi.utwente.nl/challenge7/>
This map will also be shown on the projector! If the map gets too cluttered to see your own location, you can click on your entry in the table to only show your own location. You can disable the ‘per group’ view by clicking on the table header. The map view can be enlarged by clicking on it.
- The provided code already contains the positions of all known access points (Look at the included maps

and positions in `utils.java` in the `Utils` package or `positions.cpp` in the `utils` folder). Look at the provided *SimpleLocationFinder* for a ‘working’ implementation.

3 Hints for getting started

The code we give you already works, albeit very crudely: it simply puts your location at the first measured access point that is on the list of known locations. While this is good enough to at least place your location on the map it is not very accurate, or logical. A simple improvement would be to pick the strongest access point on the list.

4 Submission and grading

As usual, your code and a report of at most 1 page have to be handed in via Blackboard. Please provide your code as a zip archive and the report in pdf format. Do not forget to write your names and student numbers in the report!

Furthermore, you have to demonstrate your system to a student assistant. In order to quantify your performance, you will be asked to demonstrate the accuracy of your system at a few designated places, where the error in your reported position will be noted.

5 Learning objectives

- Understanding localization algorithms.
- Understanding the influence of the environment on wireless signal strength.