**Conditions that trigger data point collection for a trip.**

**Definitions :-**

* **distance\_filter -** Distance interval at which data points are collected.
* **accuracy -** GPS latitude and longitude accuracy (More accurate the position, lesser the value)
* **speed -** Speed of the vehicle
* **action\_type -** Defined by motion detection classes on iOS and Android, eg. In\_vehicle, bicycle, walking, still etc.
* **cooldown mode -**  Is set to active if time gap < 300 sec form last beacon scan timeout, where beacon/car was not found.

**References** :-

* <https://github.com/ua108/ios_shared_pod/blob/develop/Documentation/DocUnderRapidCircle/AnalyseLocation%26MotionFlowChartInBackGroundTripListenerpdf.pdf>

**Process details :-**

**1)** The initial and default **distance\_filter** is **1000 meters** when the app starts.

**2)** The app constantly keeps detecting motion change or GPS/location change. If the distance\_filter is exactly **1000 meters**, no ongoing trip is found, and when we receive one GPS location or activity(in\_vehicle, bicycle), we deduct **300m** from the distance\_filter.

**3)** If there is a motion detection or GPS change, the app then checks if GPS **accuracy** (in latitude, longitude. So more accurate the position is, lesser the value

**4)** If distance\_filter is 1000m and trip is not started then the following calculation takes place.

**5)** The minimum distance calculated based on various conditions as follows :-

|  |  |
| --- | --- |
| **Condition** | **Distance\_filter Value** |
| **5.1)** action \_type is **in\_vehicle** | **Distance\_filter = distance\_filter - 100m** |
| **5.2)** Action\_type is **in\_vehicle**, accuracy < **50m** and speed>=**0m/s** | **Distance\_filter = distance\_filter - 250m** |
| **5.3)** Action\_type is **in\_vehicle**, accuracy **100m** and speed>=**8m/s** | **Distance\_filter = distance\_filter - 350m** |
| **5.4)** Action\_type is **in\_vehicle**, accuracy < **50m,** speed>=**10m/s** | If **NOT** in cooldown mode then **Distance\_filter = distance\_filter- 400m.** If in cooldown mode then **Distance\_filter = distance\_filter + 1500m.** |

**6)** The app then starts scanning for a beacon only when the below conditions are satisfied for above point **5.4:**

|  |
| --- |
| **Condition** |
| **6.1)** action \_type is **in\_vehicle** |
| **6.2)** Accuracy < 50 |
| **6.3)** Speed >= 10 m/s |
| **6.4)** Device is not in cool down mode |
| **6.5)** No trip is ongoing |

**7)** Once the correct beacon is found with the current vehicle/driver profile and **distance\_filter <100m eg. 90m**, the app starts to record the trip.

**8)** The first set of data points is collected when location is detected after first becon discovery and connection.

**9)** Thereafter, every **90m** (our calculated distance\_filter), the app requests for location and other data points to be collected.

**10**) The distance\_filter will be recalculated after every location request. Depending upon the above parameters of activity\_type, speed, accuracy some distance will be added or deducted from the distance\_filter value. If the distance\_filter value reaches upto 0m, then the data point collection will depend upon the Android/iOS platform's frequency of receiving the location changes. It can vary from 1-2 seconds by default or can be configured as well.

For Android, current configuration is to receive location update 1.002 seconds if the distance\_filter is 0m.

On iOS, [from practical experience](https://stackoverflow.com/questions/49040825/shorter-time-interval-for-location-updates), we can expect one data point (GPS latitude, longitude) per seconds with distance filter 0 when vehicle is moving (activity type is *in\_vehicle*). There is not any direct way to set the location update frequency to 1 sample or 2 samples per second. It totally depends upon the distance\_filter and movement of the vehicle. ["https://stackoverflow.com/questions/17133967/how-often-does-corelocation-update#answer-17134195"](https://stackoverflow.com/questions/17133967/how-often-does-corelocation-update#answer-17134195)

**11)** Once the car halts or slows down, if the activity type is "**in\_vehicle**" or "**bicycle**" and the accuracy and speed change is detected to be lower than the above values, here after, the app starts adding to the distance\_filter in the below conditions

* Distance\_filter calculation based on activty\_type changes (**in\_vehicle**, **bicycle**, **running** or **walking**)

|  |  |
| --- | --- |
| **Condition** | **Distance\_filter Value** |
| **10.1)** action \_type is “**bicycle”** | Distance\_filter = distance\_filter +100m |
| **10.2)** Confidence **100%** & motion type is not “**in\_vehicle**” | Distance\_filter = distance\_filter +150m |
| **10.3)**action\_type is not **“in\_vehicle”** | Distance\_filter = distance\_filter +50m |

* Distance\_filter calculation based on GPS location changes.

|  |  |
| --- | --- |
| **Condition** | **Distance\_filter Value** |
| **10.4)** action\_type is “**in\_vehicle”,** location accuracy **< 50m,** speed **>= 10m/s** andCoolDown mode is **active** | Distance\_filter = distance\_filter +1000m |
| **10.5)** action\_type is not “**in\_vehicle”** | Distance\_filter = distance\_filter +100m |

**12)** Once the distance\_filter reaches upto **1000m/1km (950m for iOS)**, the trip is stopped, and post processing is done to create a request body for trip upload.

**13)** The trip might also be stopped if there is low battery signal (Android <15%, iOS <20%)or GPS is disabled.

**Conclusion: - The data point collection depends on the below conditions :-**

|  |  |
| --- | --- |
| **Condition** | **Result** |
| * If the trip is **NOT** on-going, we still receive the location even if the distance\_filter is anywhere between 100m-1000m | Data points are **NOT** saved or recorded |
| * If the trip is **NOT** on-going and calculated distance\_filter is between **0-100m** along with the above mentioned parameter requirements points from section **6.1** - **6.5** | Start a new trip and collect data points based on calculated distance\_filter eg. If the distance\_filter is **90m**, then collect a new set of data points every **90m**. |
| * If trip **is** on-going | Collect data points depending on the distance filter value between **0-1000m**. If the distance\_filter value is **0m**, then data collection depends upon the platform's frequency of recieving the location change updates. Once the distance\_filter value reaches 1000m (950m for iOS), stop the trip. |