Progress report

Date:2019-05-05

- Filter suitable cars
- Simulator environment (edited version)
- Division of the task
- New task

Task List

~~1.arrange 10 random cars in the more concise map~~

~~2.compute the minimum distance between the sender and the ten cars~~

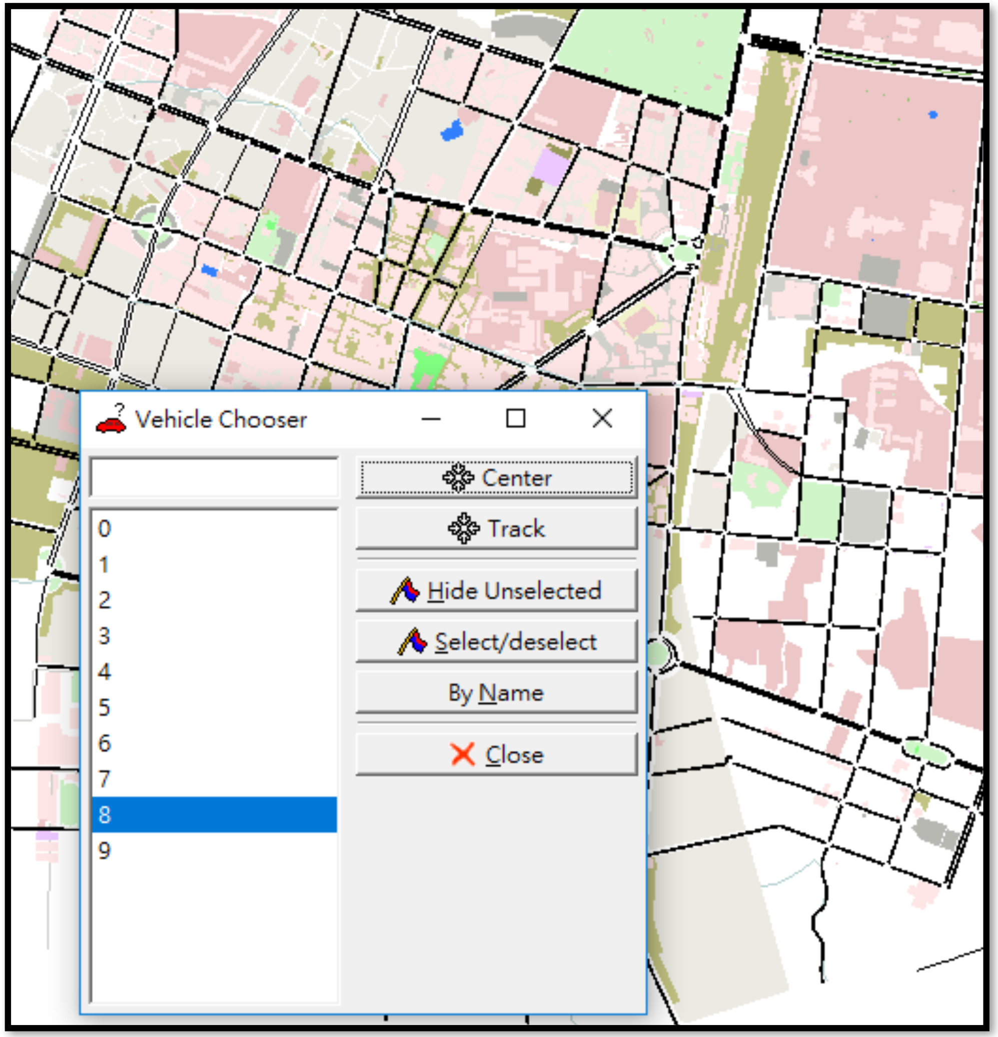
~~3.Dispatch the selected car to the sender~~

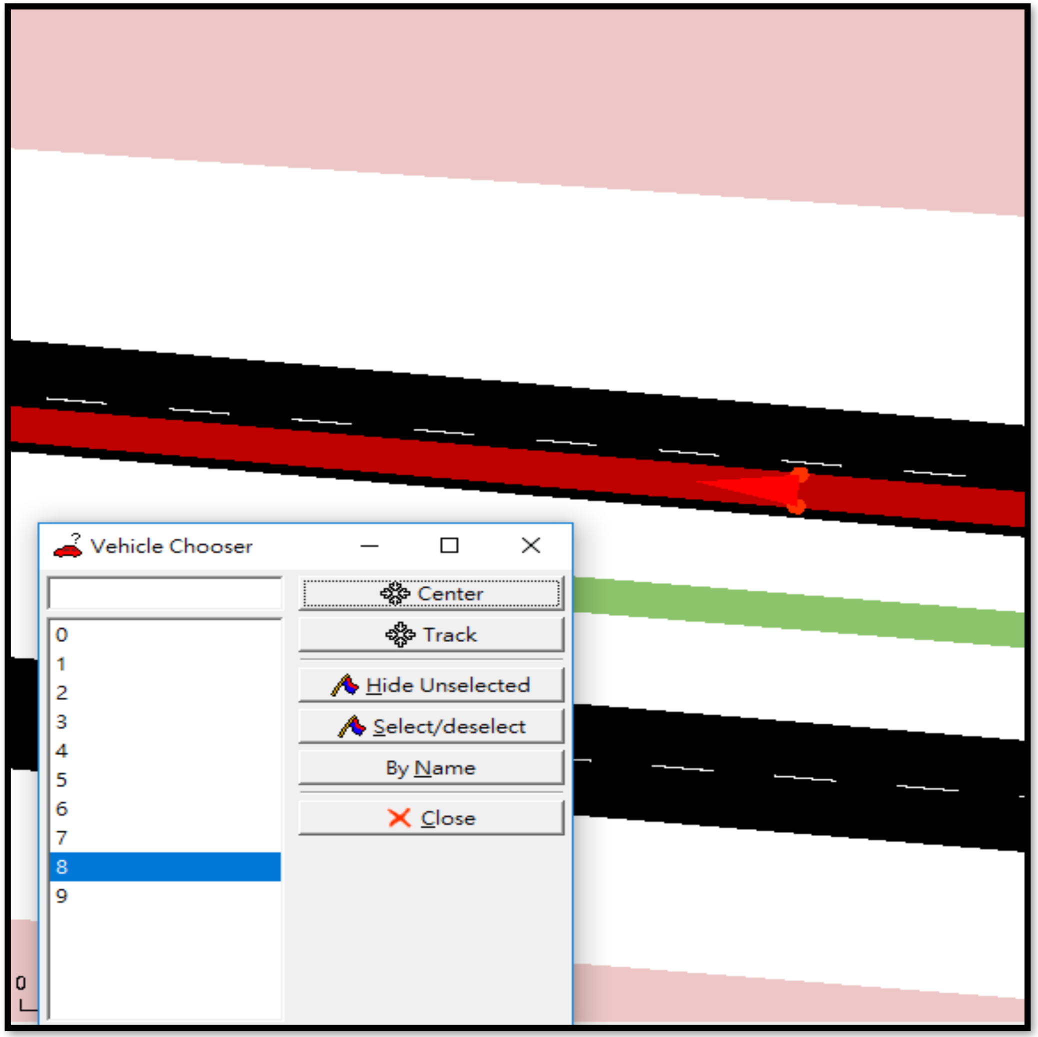
~~4.Add **travel-time estimation** function~~ and notification to the user connecting the Android

5.Try to receive the geo-position from user's request and save it in a appropriate datatype (ex. arrayList)

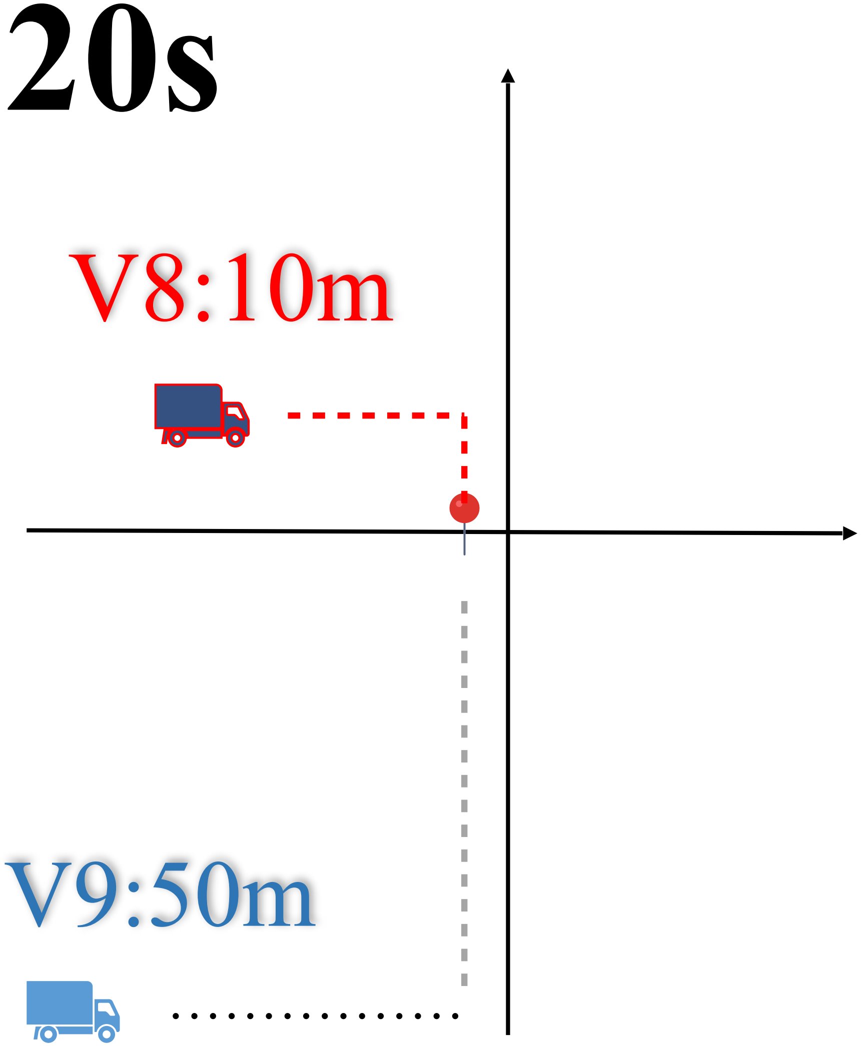
6.Report the related data to the webserver (eg. the geo-position of the car)

Arrange 10 random cars in the more concise map





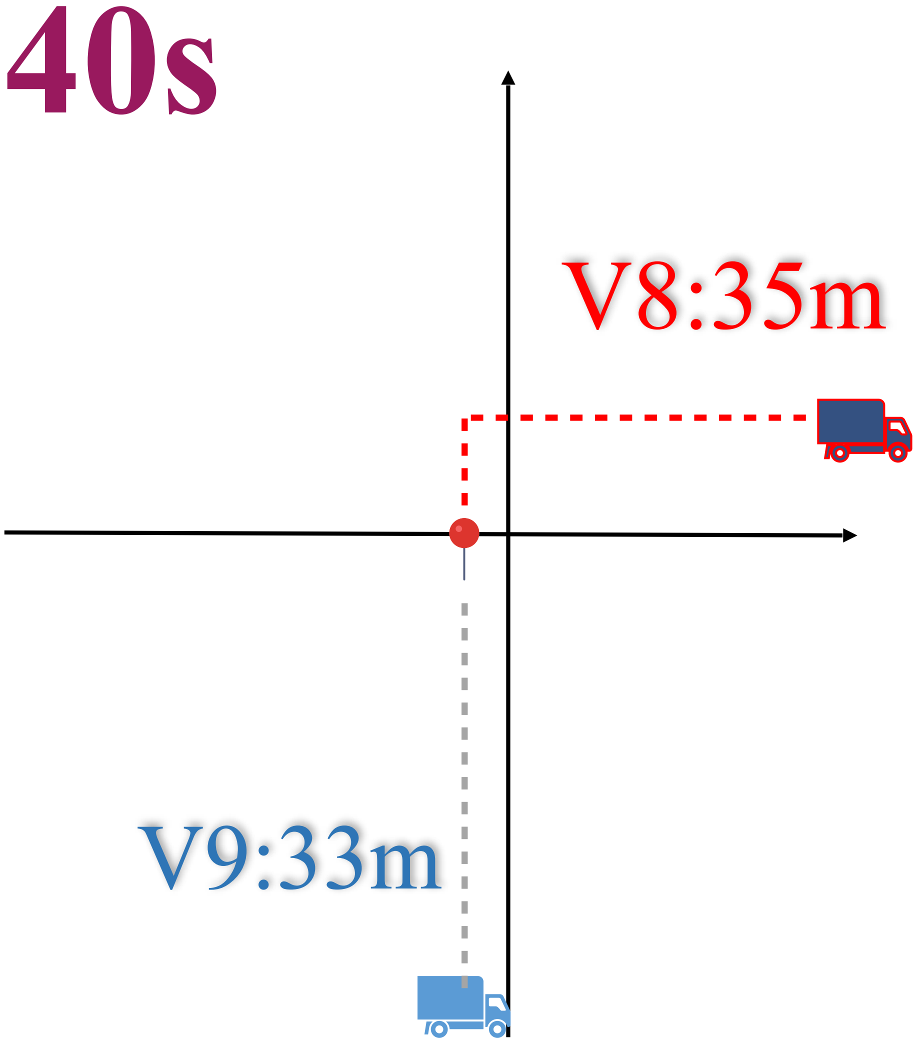
Compute the distance



20s

V8:10m

V9:50m



40s

V8:35m

V9:33m

Implementation results

1s

- current v8_edgeID is: 313194185#3
- current v8_position x: 4522.27833916253 y: 1920.8552667432366
- current distance between v8 to sender is:**5075**.269336728843
- current distance between v9 to sender is:**4736**.283500394948

2s

- current v8_edgeID is: 313194185#3
- current v8_position x: 4519.158030583849 y: 1921.896613339231
- current distance between v8 to sender is: **5073**.342501706793
- current distance between v9 to sender is: **4739**.520232601273

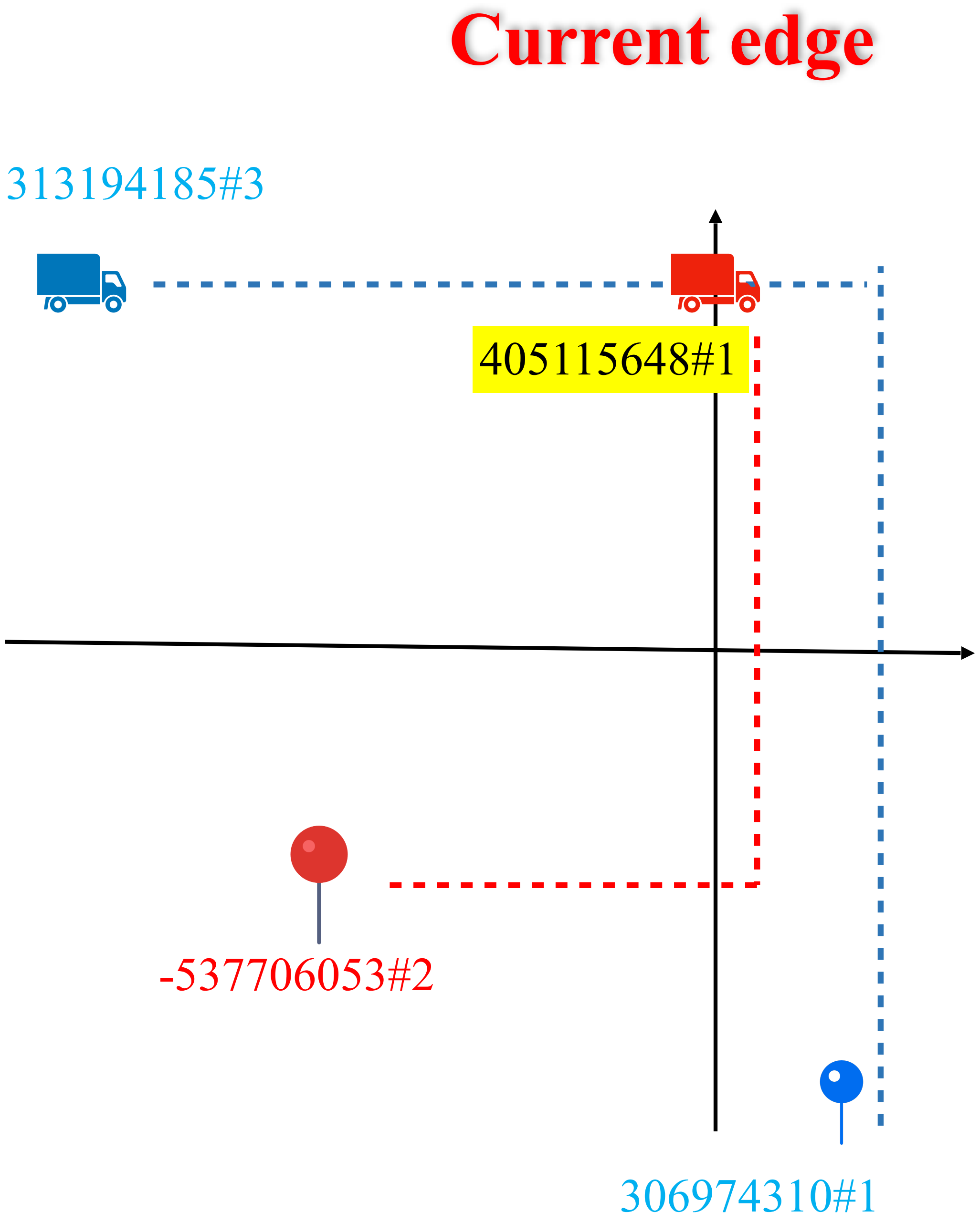
3s

- current v8_edgeID is: 313194185#3
- current v8_position x: 4514.207321846696 y: 1923.548822907995
- current distance between v8 to sender is: **5070**.288245151509
- current distance between v9 to sender is: **4744**.725440100539

Received the request of the sender at 60.0s

- if((v9toSenderDistance< v8toSenderDistance) && timeSeconds==60.0) {
- System.out.println("we dispath v9 to the sender address!”);}

- else if(v9toSenderDistance > v8toSenderDistance && (timeSeconds==60.0))
- {System.out.println("we dispath v8 to the sender address!");}



defaultRouteList:[313194185#3, 313194185#6, 313194185#11, 313194185#14, 496257370#0, 405115648#1,
313194390#0, 675775398, 496257372, -315702598#2, -228022792#6, -228022792#2, 72871344, 72871329#2, 72871329#3, 27067581, 228022808#0, 228022808#2, 228022808#4, 228022808#6, 306974310#0, 306974310#1]

changedRouteList:[313194185#3, 313194185#6, 313194185#11, 313194185#14, 496257370#0,
405115648#1,
-307096543#5, -537706053#4, -537706053#2]

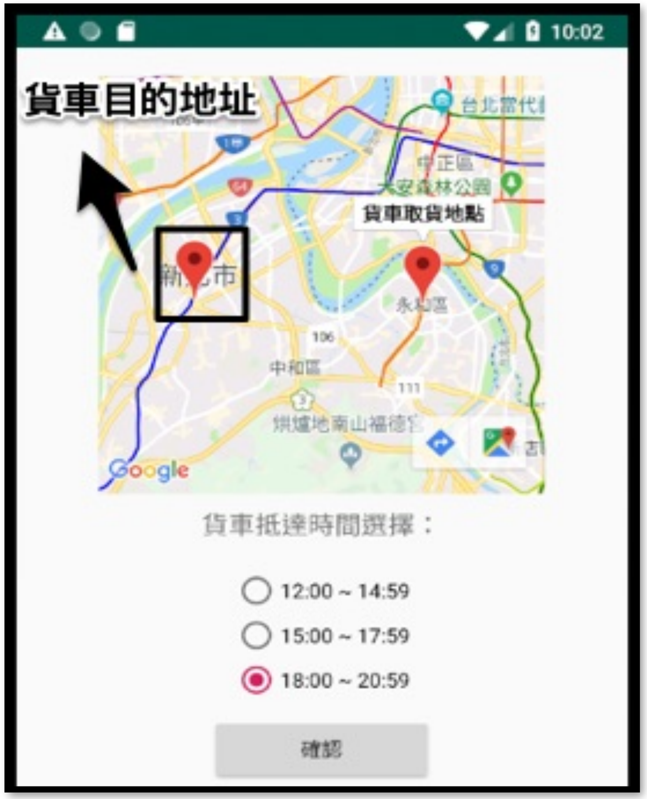
We need **75.56 s** from current edge to sender address (**V_{max}=20 m/s**)

The condition to filter the suitable cars

1.shortest distance

2.containers number (Transporting? Lower than container capacity?)

3.the selection of time Interval



貨車目的地址

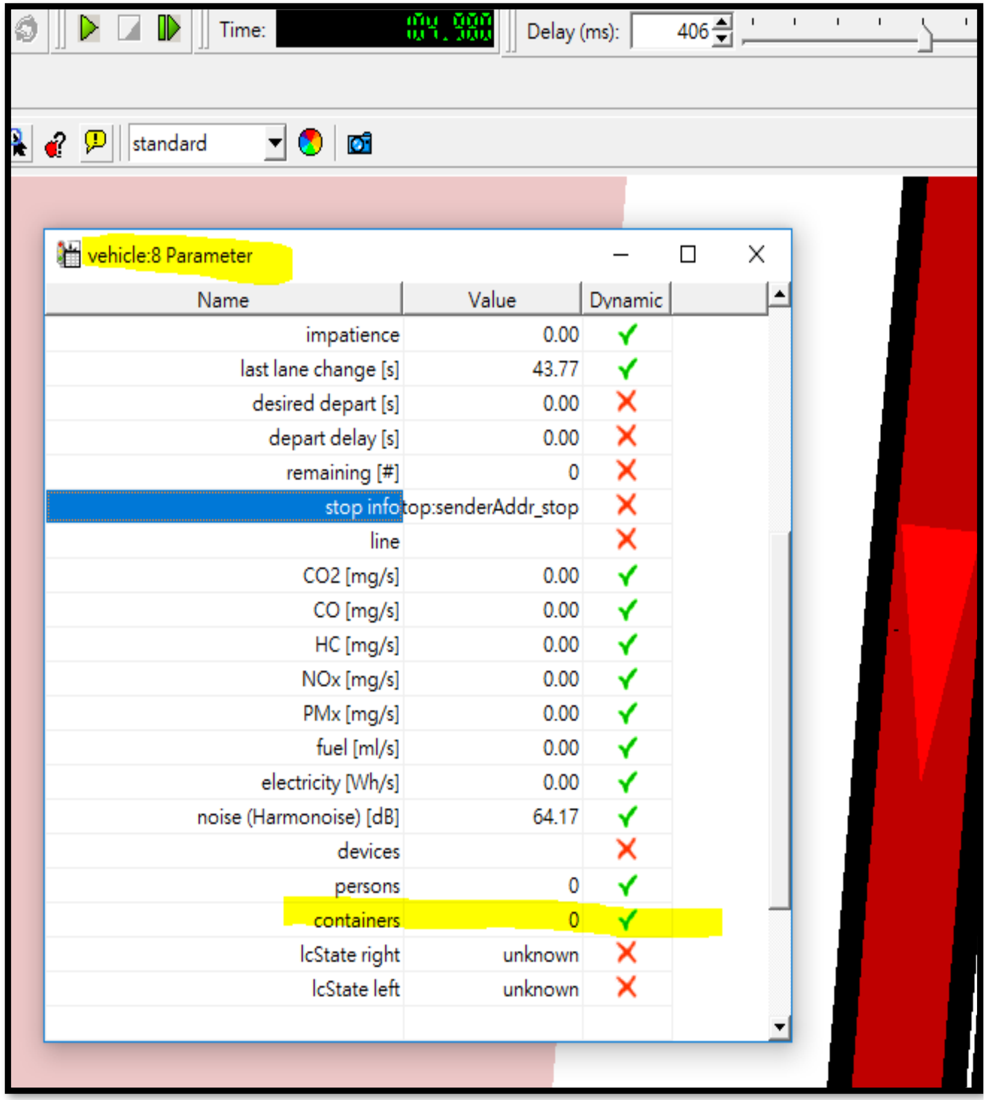
貨車抵達時間選擇：

○ 12:00 ~ 14:59

○ 15:00 ~ 17:59

● 18:00 ~ 20:59

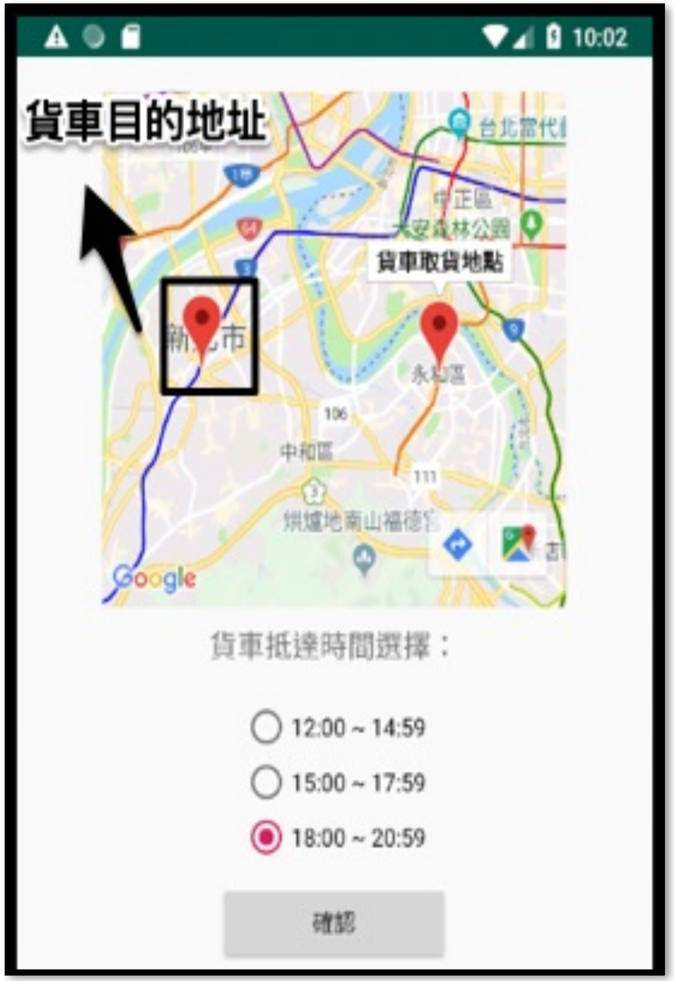
確認



vehicle:8 Parameter

Name	Value	Dynamic
impatience	0.00	✓
last lane change [s]	43.77	✓
desired depart [s]	0.00	✗
depart delay [s]	0.00	✗
remaining [#]	0	✗
stop info	top:senderAddr_stop	✗
line		✗
CO2 [mg/s]	0.00	✓
CO [mg/s]	0.00	✓
HC [mg/s]	0.00	✓
NOx [mg/s]	0.00	✓
PMx [mg/s]	0.00	✓
fuel [ml/s]	0.00	✓
electricity [Wh/s]	0.00	✓
noise (Harmonoise) [dB]	64.17	✓
devices		✗
persons	0	✓
containers	0	✓
lcState right	unknown	✗
lcState left	unknown	✗

The selection of time Interval



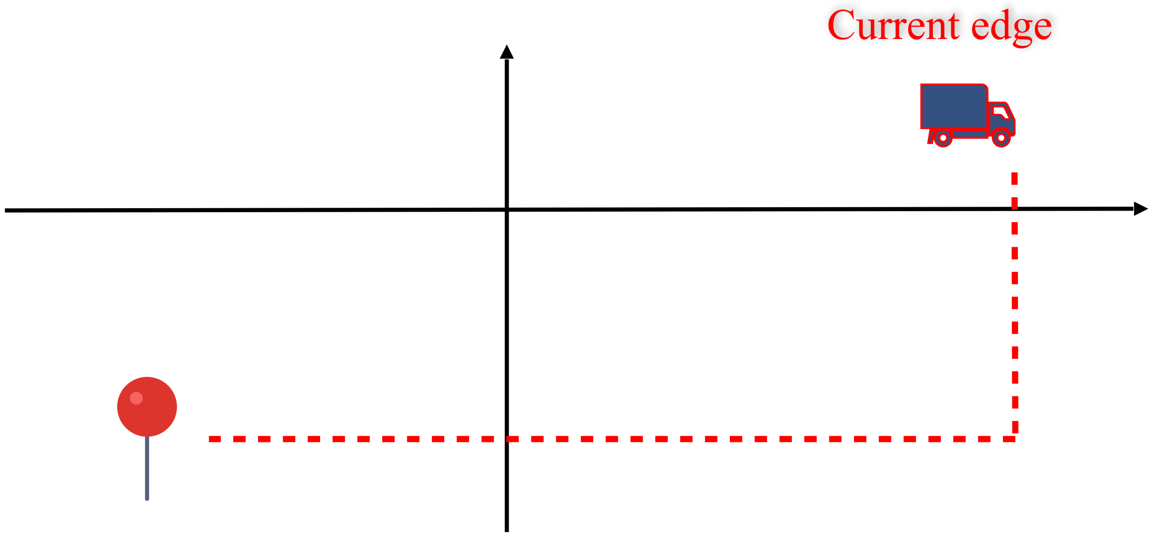
貨車抵達時間選擇：

○ 12:00 ~ 14:59
○ 15:00 ~ 17:59
● 18:00 ~ 20:59

12:00~14:59

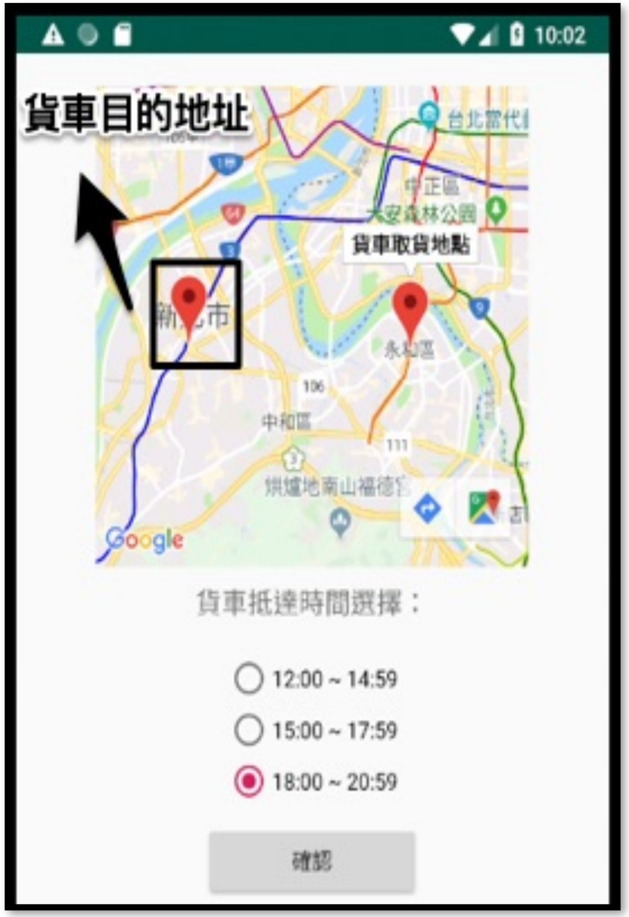
- Vmax=20 m/s
- lower bound:1hr58mins (1*60+58)*60*Vmax=141600 [m]
- upper bound:4hr57min (4*60+57)*60*Vmax=356400 [m]

141600 < The distance < 356400



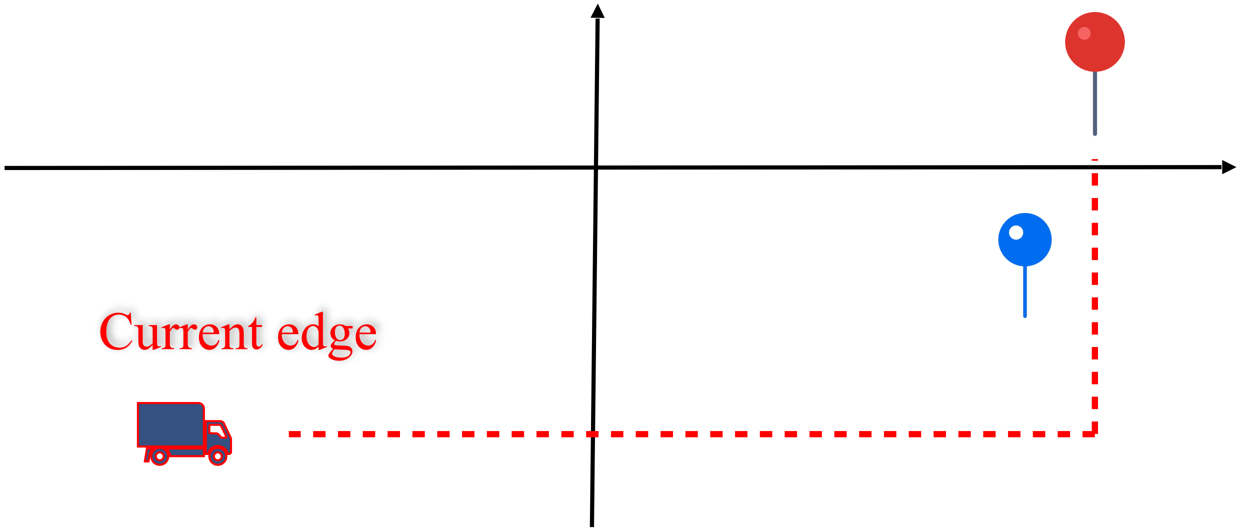
Waiting time

- Current time:10:02
- The sender selected 18:00~20:59
- We assume the car would arrive to the sender’s address at **15:00.**
- We would make the car waiting at the place nearby the destination from **15:00~18:00**

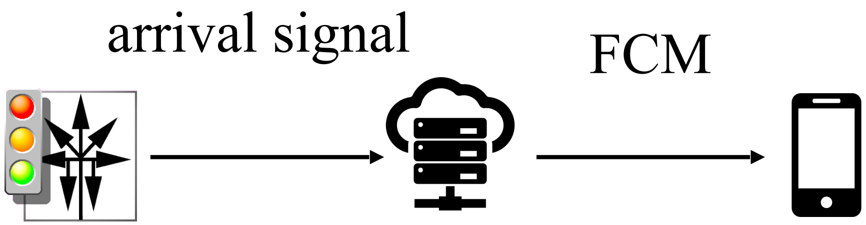


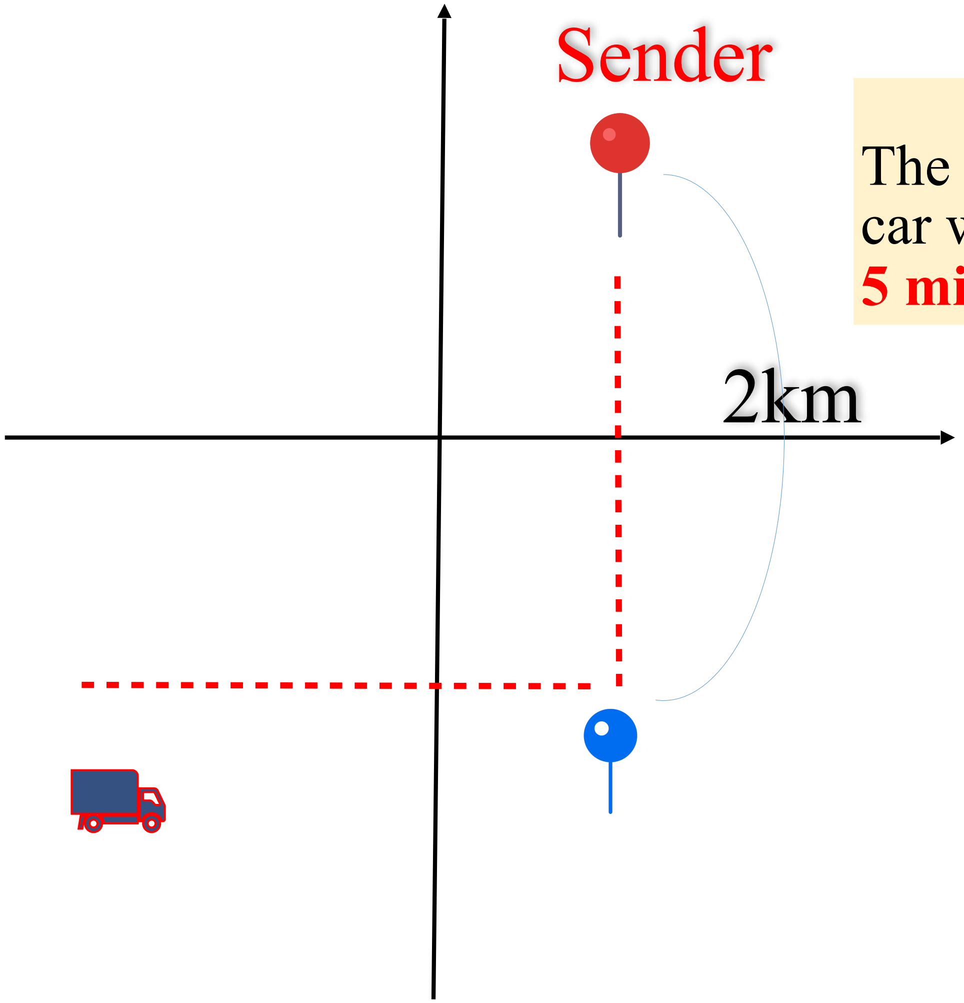
貨車抵達時間選擇：

○ 12:00 ~ 14:59
○ 15:00 ~ 17:59
● 18:00 ~ 20:59

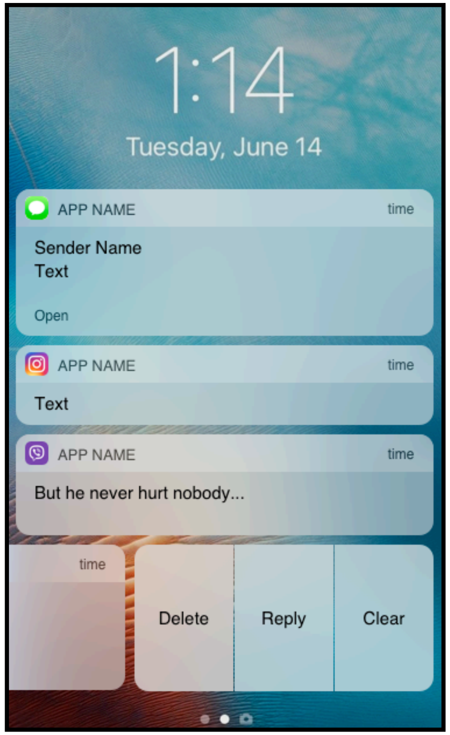


Notification to the sender within 2km





The sender would receive the message that the car would arrive to the destination within **5 minutes (or position)**



1:14

Tuesday, June 14

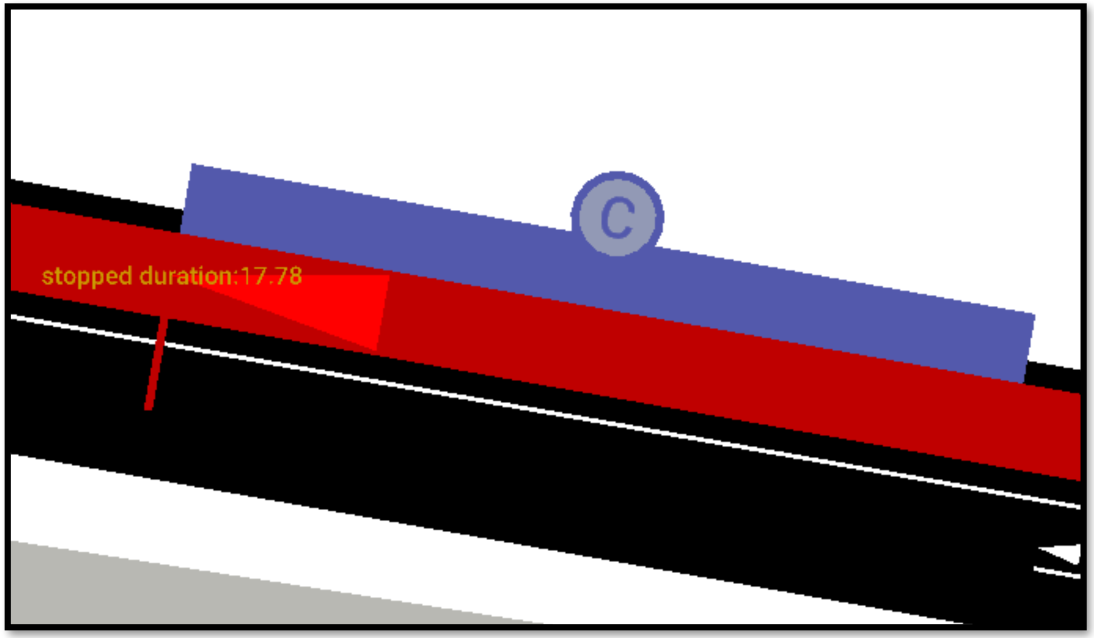
APP NAME time
Sender Name
Text
Open

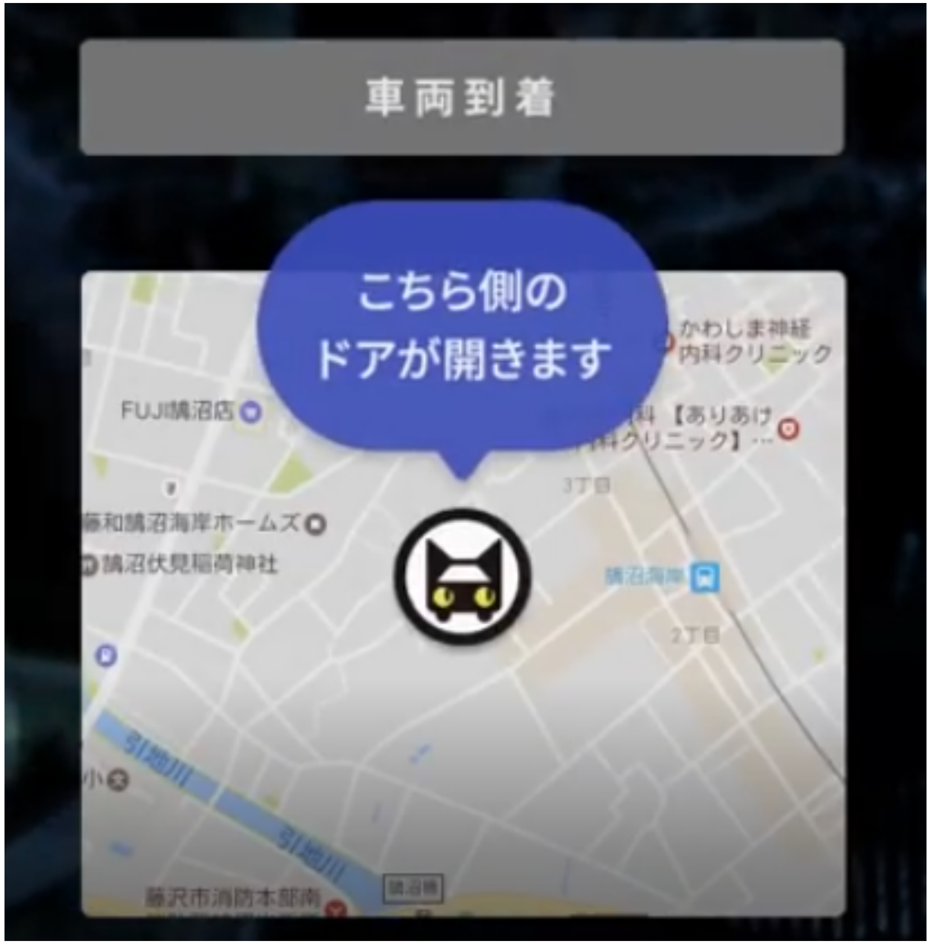
APP NAME time
Text

APP NAME time
But he never hurt nobody...

Arrive to the sender

1.到點通知，以廣告推播型式送給sender




車両到着

こちら側の
ドアが開きます

Arrive to the sender (2)

2.SUMO-server發訊息給receiver確認收貨時間

(讓sender有預設時間or最短時間)

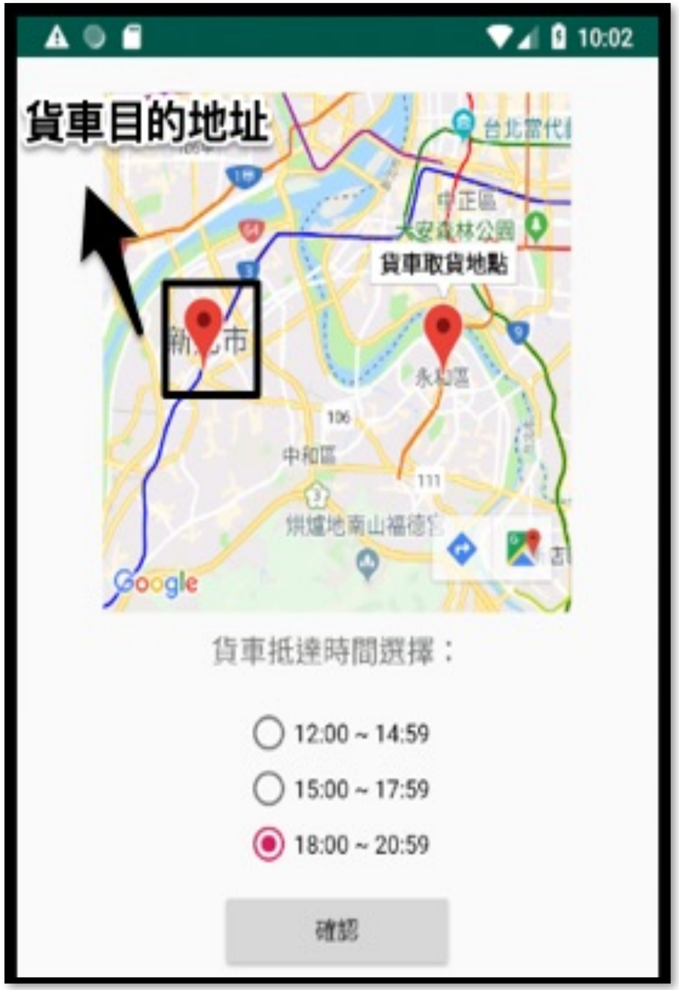
3.SUMO-server接收receiver傳回的確認信息

4.SUMO安排路徑與設置在receiver地址停下

5.等待sender按下trigger鍵，以確認上貨完畢

6.結束目前stop stage //用resume功能

7.車子從sender出發，前往receiver


10:02

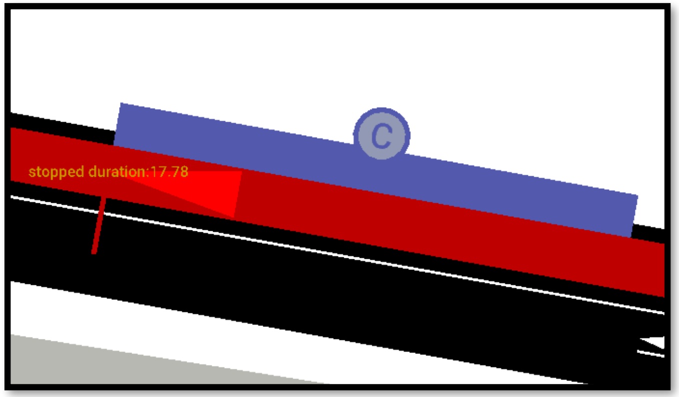
貨車目的地址

貨車抵達時間選擇：

○ 12:00 ~ 14:59

○ 15:00 ~ 17:59

● 18:00 ~ 20:59

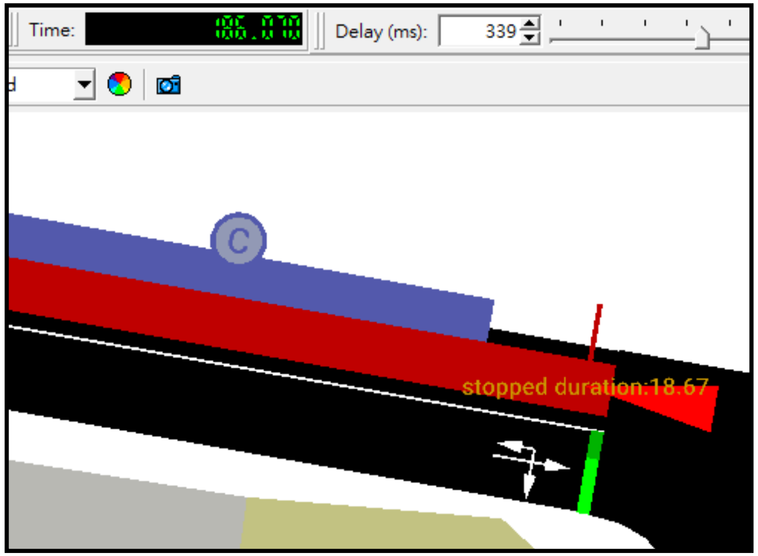

stopped duration:17.78

Cancel the stop stage: Resume

Vehicle.resume("8")

196s

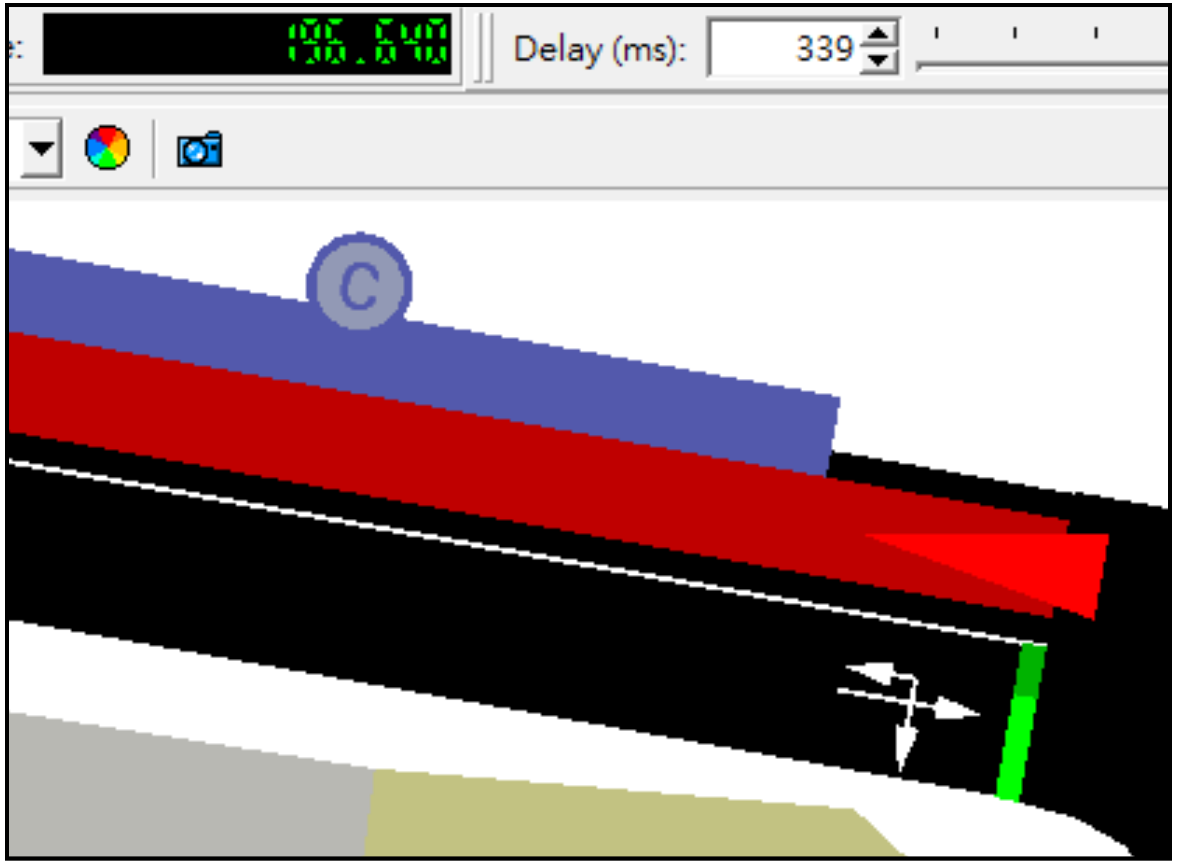
186s



Time: 186.070

Delay (ms): 339

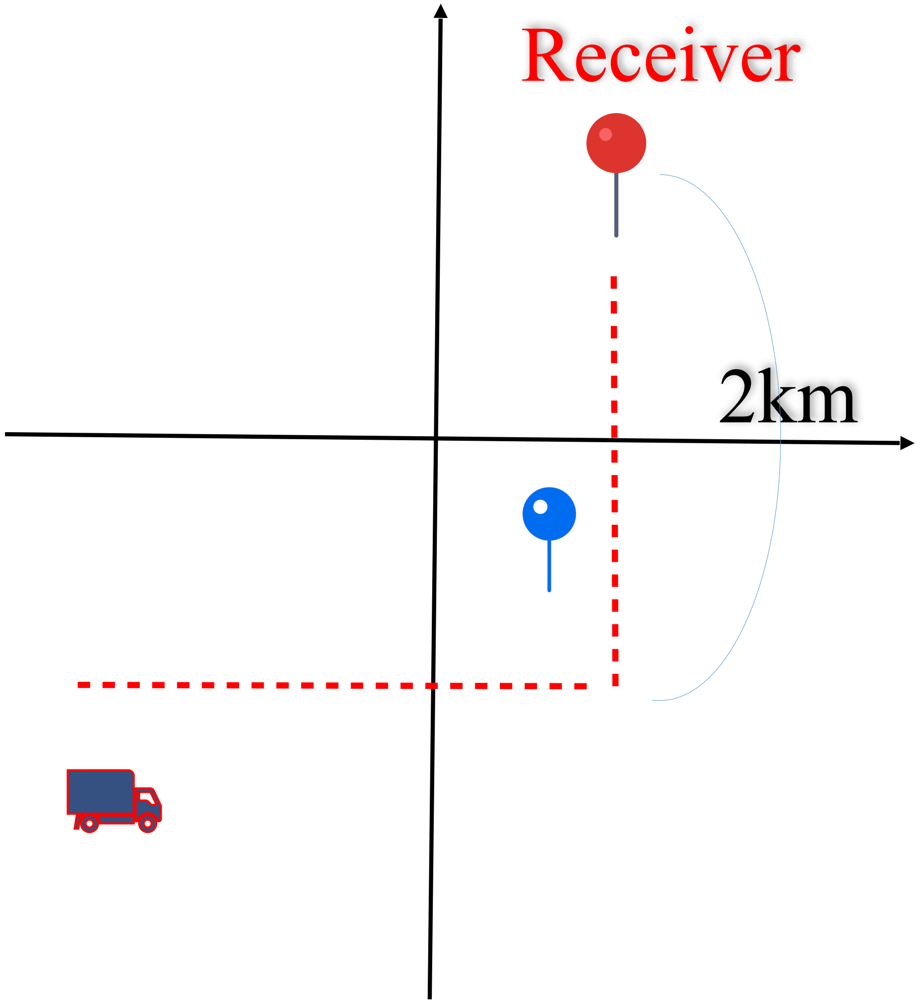
stopped duration:18.67

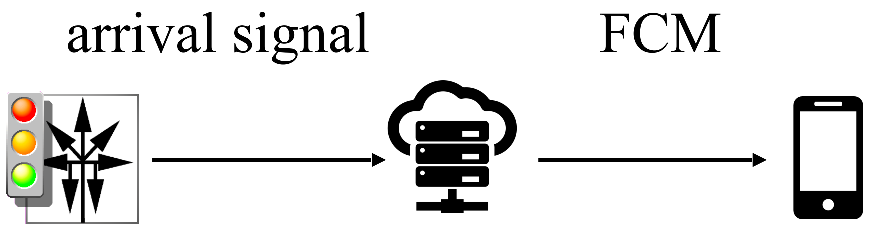


196.640

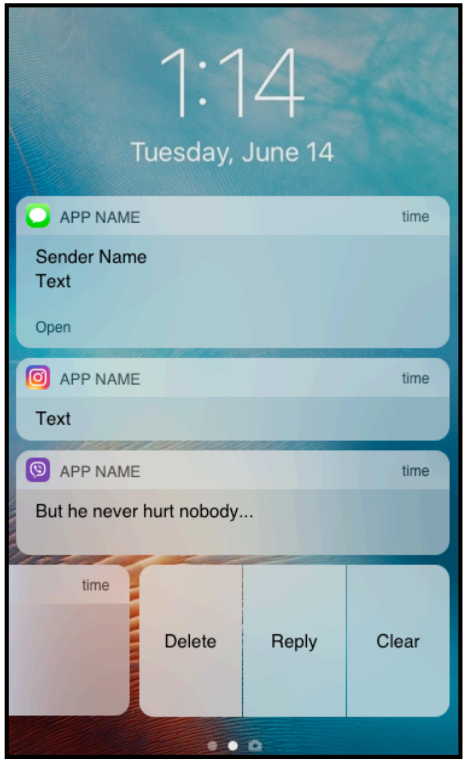
Delay (ms): 339

Notification to the receiver within 2km





The receiver would receive the message that the car would arrive to the destination within **5 minutes**



1:14

Tuesday, June 14

APP NAME time

Sender Name
Text

Open

APP NAME time

Text

APP NAME time

But he never hurt nobody...

time

Delete Reply Clear

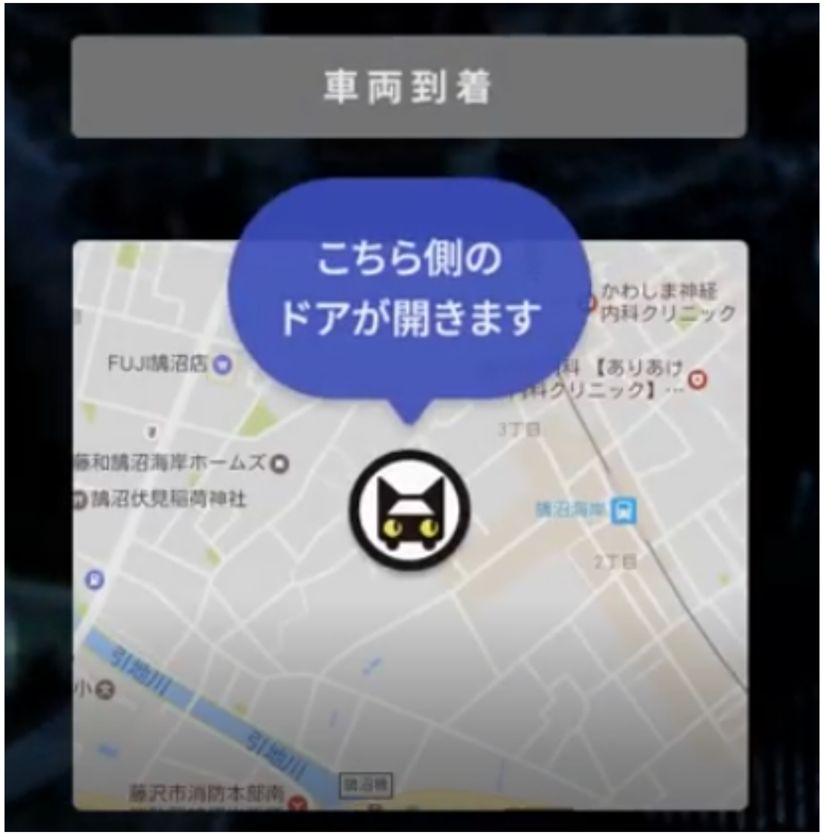
Arrive to the receiver’s address

1.到點通知，以廣告推播型式送給 receiver

2.等待receiver按下trigger鍵，確認收貨

3.結束目前stop Stage

4.車子從receiver離開，前往火車站 (停車區)



車両到着

こちら側の
ドアが開きます

分工部分

布

- 推播通知
- 手機訂單查詢
- Android UI
- webserver
- 連線功能

慷

- 車輛派遣
- 路線規劃
- 上下卸貨
- SUMO-API撰寫
- 地圖資料處理

New Task List

~~1.arrange 10 random cars in the more concise map~~

~~2.compute the minimum distance between the sender and the ten cars~~

~~3.Dispatch the selected car to the sender~~

~~4.Add **travel-time estimation** function~~ and notification to the user connecting the Android

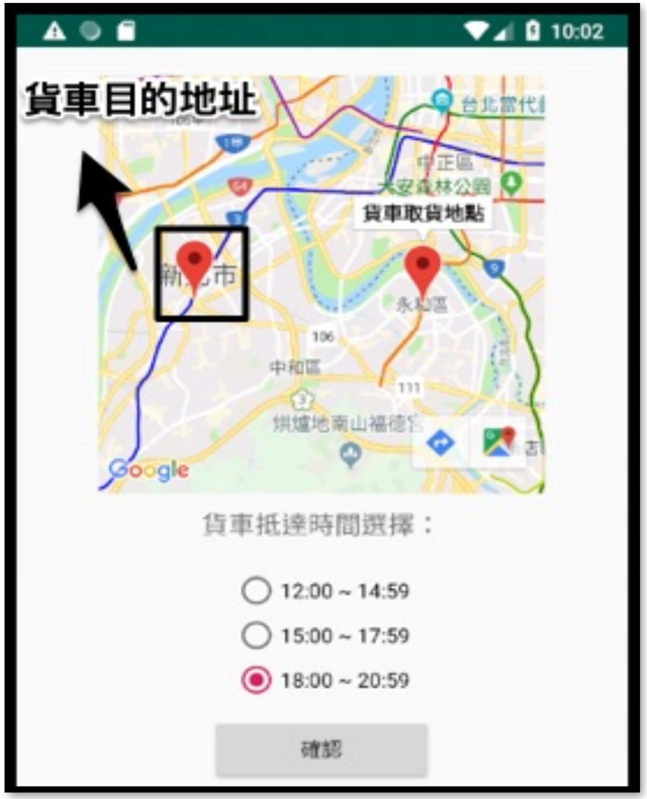
5.Try to receive the geo-position from user’s request and save it in a appropriate datatype (ex. arrayList)

6.Report the related data to the webserver (eg. the geo-position of the car)

7.Add the scenario triggered by the sender

8.Add parameters of container numbers and container capacity

9.Add time-interval selection and integrate the real time-counting function



貨車目的地址

貨車抵達時間選擇：

○ 12:00 ~ 14:59
○ 15:00 ~ 17:59
● 18:00 ~ 20:59

確認