文檔

2019-04-13(六)

接收手機端需求

- 1.Client首先會接收到request from android
- 2.這個request包括
  - 寄件地址+收件地址(以經緯度型態表示)
  - 貨車抵達收件人地址的time interval挑選
  - 訂貨單號跟車子的關係(一對一 V.S 多對一)
- 3.Client會把傳入的source geo-position, desitnation geo-position 轉成Sender Address edgeID, Receiver Address edgeID, Sender gui-position and Receiver gui-position
- 4.Client也會對現況所有車子做getPosition()與getRoadID(), 取得每一台車的gui-position與目前位置edgeID

計算現有車輛與Sender之間距離

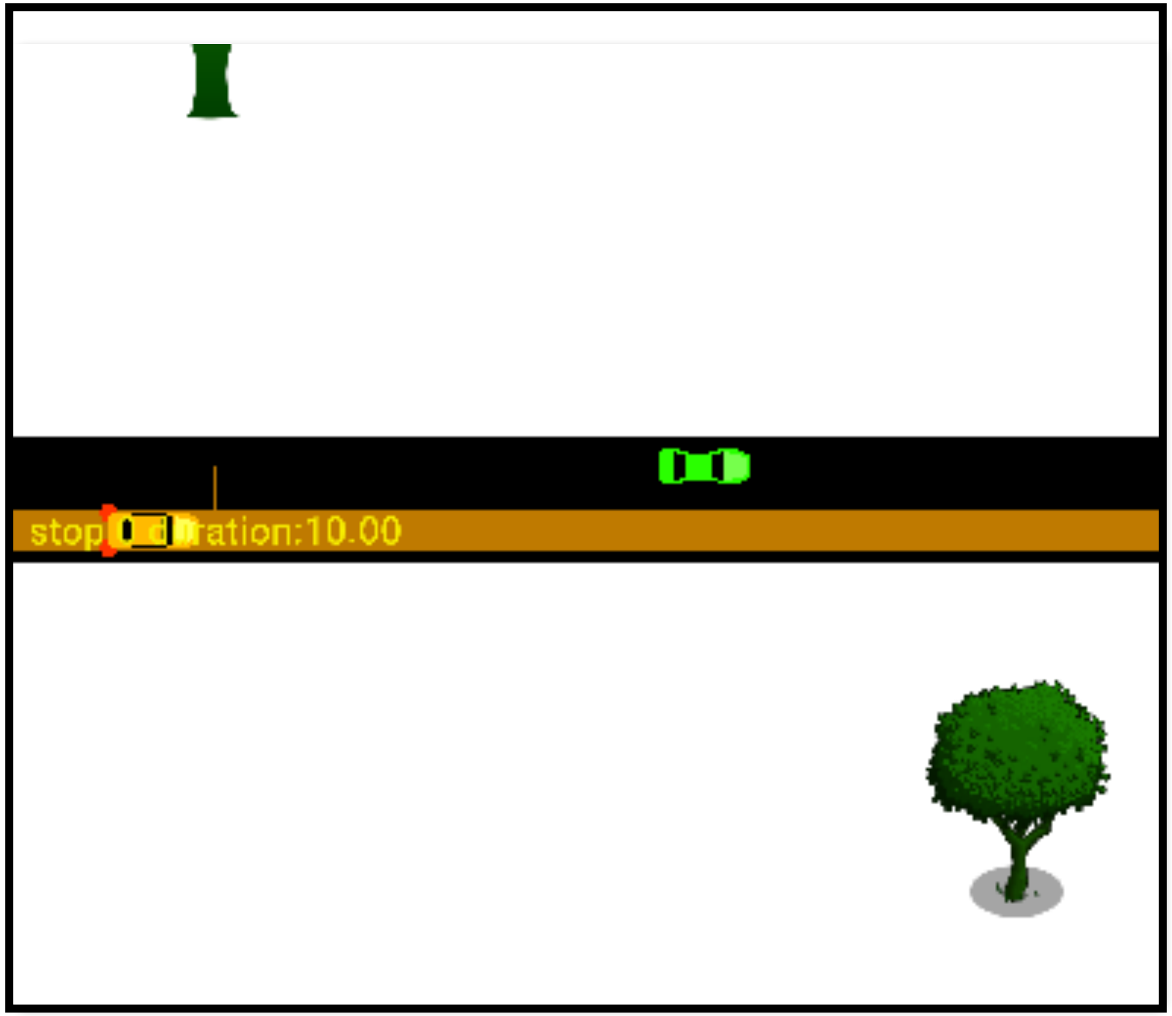
- 1.剛剛把所有的車輛資訊，提取each vehicle gui-posion(假如10台車)，那計算這10台車(car1,car2,…car3)與sender gui-position之間的距離
- 2.利用getDistance2D(self, x1, y1, x2, y2, isGeo=False, isDriving=False) getDistance2D(double, double, double, double, boolean, boolean) -> double
- 3. distance5=min{distance1, distance2,…distance10}
- 4.那就把car5認定成配送車

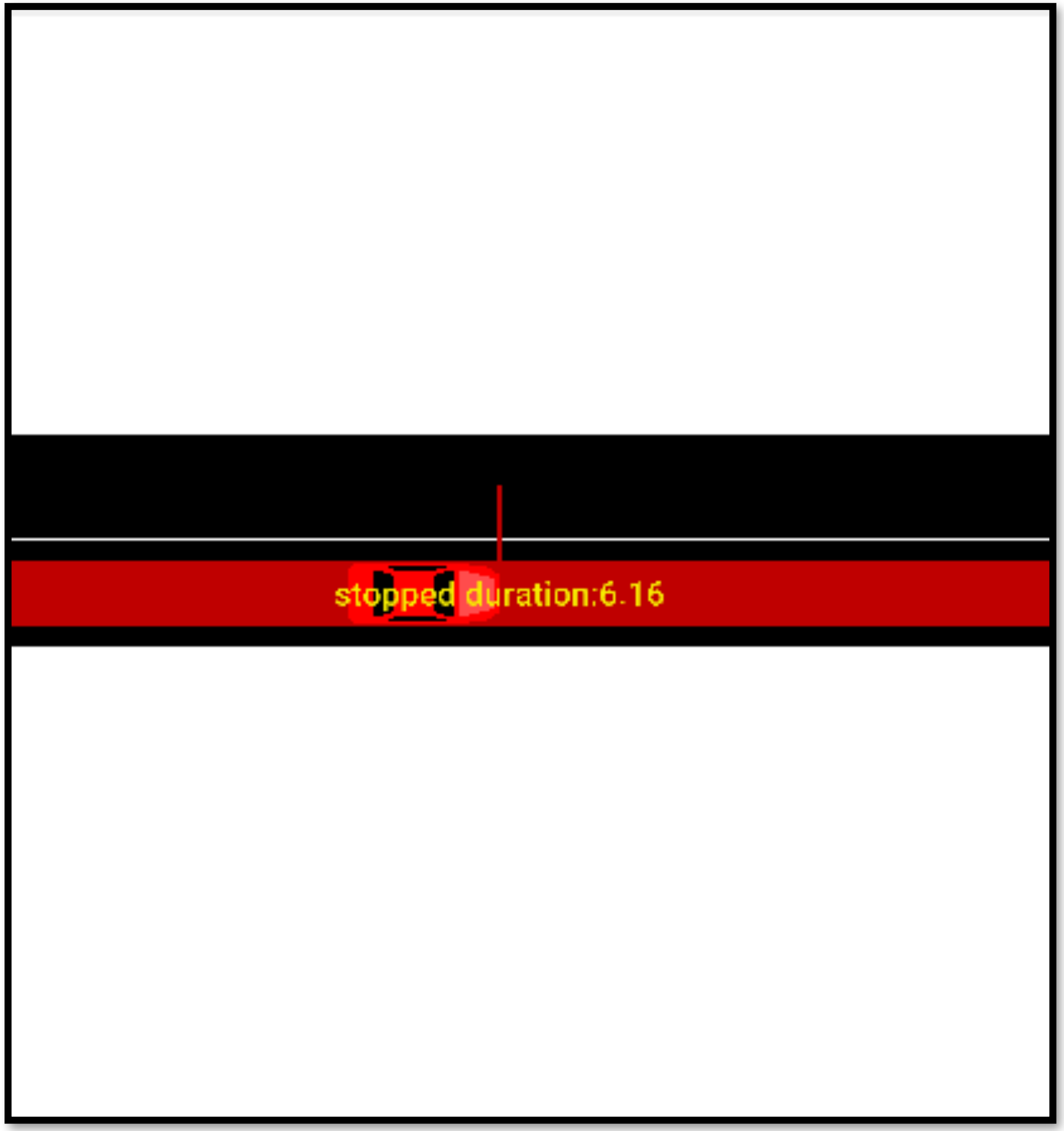
動態路徑規劃

- 1.把剛才挑選的car5設定為派遣車輛，取得car5的gui-position(x,y座標)與car5_roadID(也就是car5_edgID)
- 2.給startEdge: car5_edgID, endEdge:sender_address_edgeID
- 3.接著是動態規畫路徑，目前還沒查到ws的相關用法，所以用TraCI(python)代替表示:
  - traci.route.add("trip", ["startEdge", "endEdge"])
  - traci.vehicle.add(“newVeh”, “trip”, typeID=“reroutingType”) //增加這台車
- 4.關鍵在於如何用scripts與起始終點邊，規劃出一條route，，比較接近的是ws.RouteAdd(String routeID, List<String> edges)

前往sender address

- 1.從car5目前所在地，前往sender address時，要停下來
- 2.因此用到該函數**Vehicle_setStop**
- 3.public void **Vehicle_setStop**(String vehID, String edgeID, double pos, byte laneIndex, int duration, **SumoStopFlags** stopType)
- 4. Class **SumoStopFlags** //*http://sumo.sourceforge.net/javadoc/traas/de/tudresden/ws/container/SumoStopFlags.html*
- 5.注意**SumoStopFlags**有很多parameters，例如isStopped, setStoppedvalue, isContainerTriggered, setIsContainerStop
- 6.當車子到sender address時，要做isIsContainerStop()的參數調整，還有load container的調整
- 7.SumoStopFlags能用的method like **notify(),wait()**


stop duration:10.00


stopped duration:6.16

從收件者到寄件者位置

- 1.當再車子在sender停下來時，會計算stop duration，例如停10秒
- 2.再10結束後，應當安排從寄件者到收件者路徑規劃，
  - traci.route.add("trip_StoR", [“senderEdge", “ReciverEdge"])
  - traci.vehicle.add(“newVeh”, “trip_StoR”, typeID=“reroutingType”)
  - **setContainerStop()**
- 3. **setContainerStop**(self, vehID, stopID, duration=-1073741824.0, until=-1073741824.0, flags=0)
  - **setContainerStop**(string, string, double, double, integer) -> None Adds or modifies a container stop with the given parameters.
  - The duration and the until attribute are in seconds.
- 4.當arrive receiver address時，要停留一陣子，並且用SumoStopFlags的method like **notify()**外界

Geo-postion to gui-position

```
System.out.println(conn.do_job_get(Simulation.convertGeo(3414.680, 5591.166,
false ))); // 120.22115148153793,22.985924116857188
System.out.println(conn.do_job_get(Simulation.convertGeo(120.216228, 22.987473,
true ))); // 2912.904071369441,5772.349525868427
```

Before

- Lon:120.216228
- Lat:22.987473
- 用clipboard貼上的
- X:2912.41
- Y:5772.38

After

- X:2912.904071369441
- Y:5772.349525868427
- 誤差
  - X_before:2912.41
  - X_after:2912.904071369441
  - Y_before:5772.38
  - Y_after:Y:5772.349525868427

Progress report

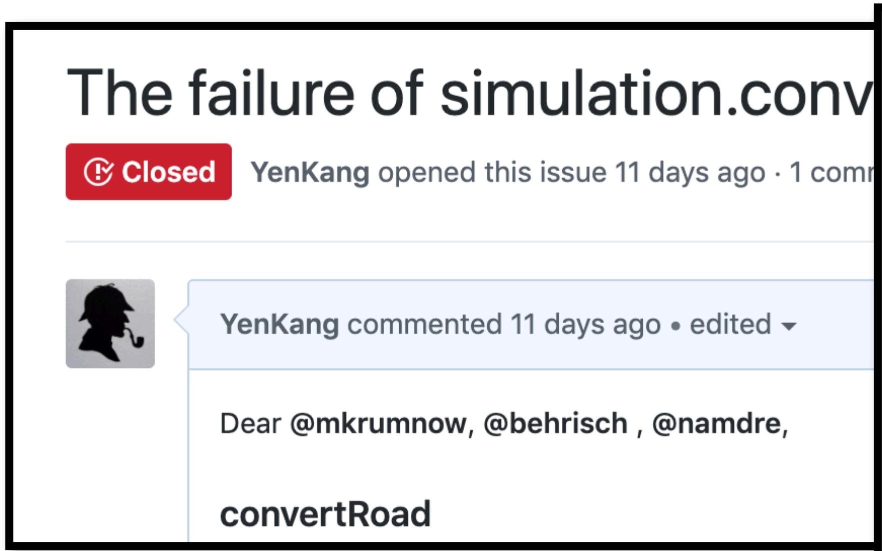
Date:2019-04-29

- **converRoad** issue fixed
- findRoute issue fixed
- changeRoute demo
- Next Step

Confirm ConvertGeo

```
convertGeo(3414.680, 5591.166, false ))

120.22021170569616,23.031769661295733
```

```
convertGeo(120.22021170569616,
23.031769661295733, true ))

3414.679998779553,5591.165999166202
```



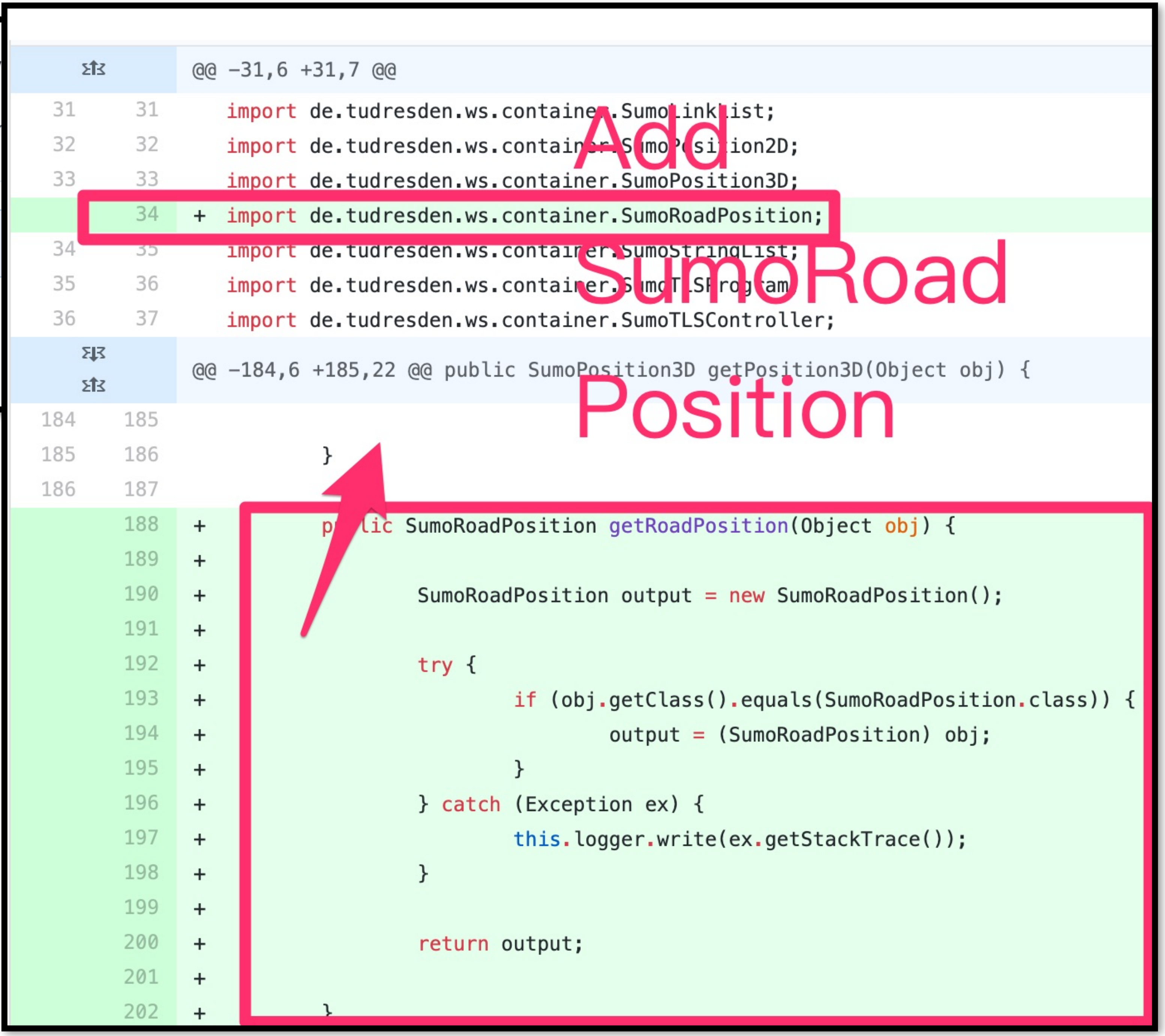
The failure of simulation.conv

Closed YenKang opened this issue 11 days ago · 1 comm

YenKang commented 11 days ago • edited

Dear @mkrumnow, @behrisch , @namdre,

convertRoad

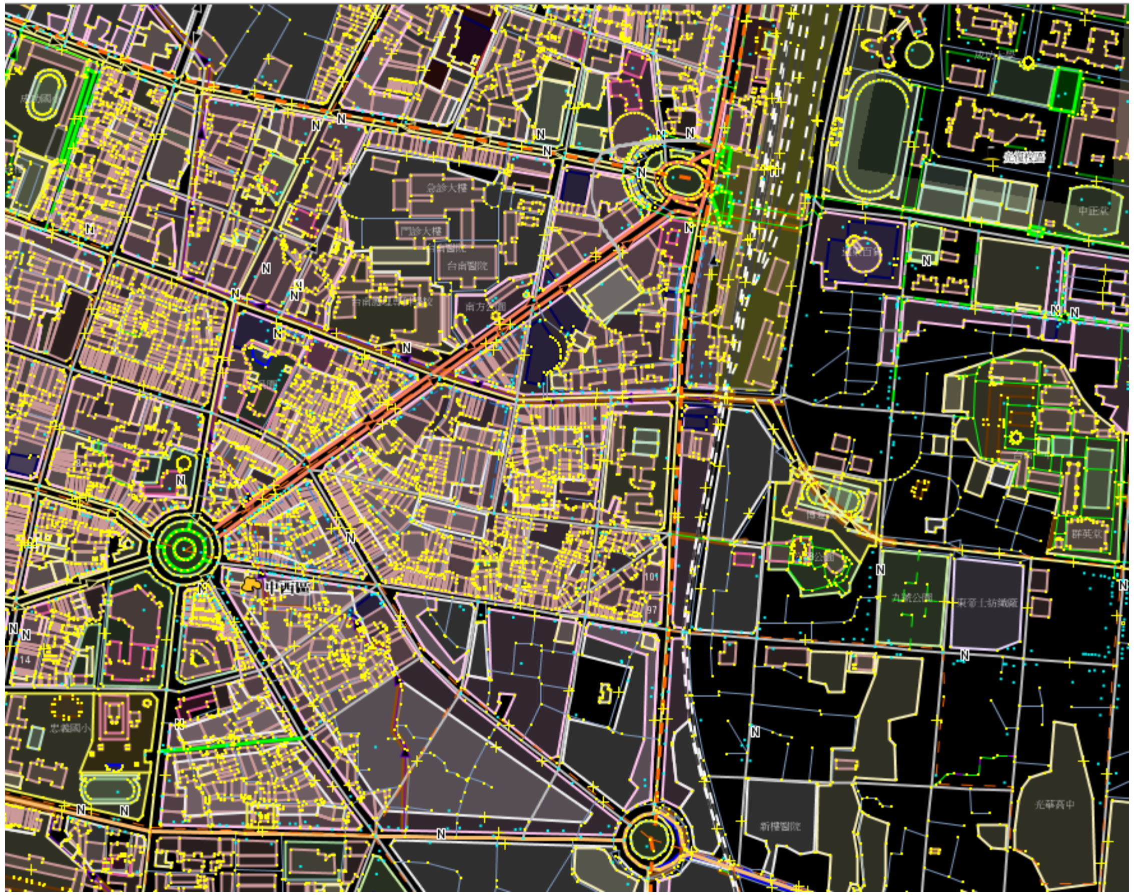


```
@@ -31,6 +31,7 @@
   import de.tudresden.ws.container.SumoLinkList;
   import de.tudresden.ws.container.SumoPosition2D;
   import de.tudresden.ws.container.SumoPosition3D;
+  import de.tudresden.ws.container.SumoRoadPosition;
   import de.tudresden.ws.container.SumoStringList;
   import de.tudresden.ws.container.SumoTLSProgram;
   import de.tudresden.ws.container.SumoTLSController;
@@ -184,6 +185,22 @@ public SumoPosition3D getPosition3D(Object obj) {

           }

+          public SumoRoadPosition getRoadPosition(Object obj) {
+
+                  SumoRoadPosition output = new SumoRoadPosition();
+
+                  try {
+                          if (obj.getClass().equals(SumoRoadPosition.class)) {
+                                  output = (SumoRoadPosition) obj;
+                          }
+                  } catch (Exception ex) {
+                          this.logger.write(ex.getStackTrace());
+                  }
+
+                  return output;
+
+          }
```

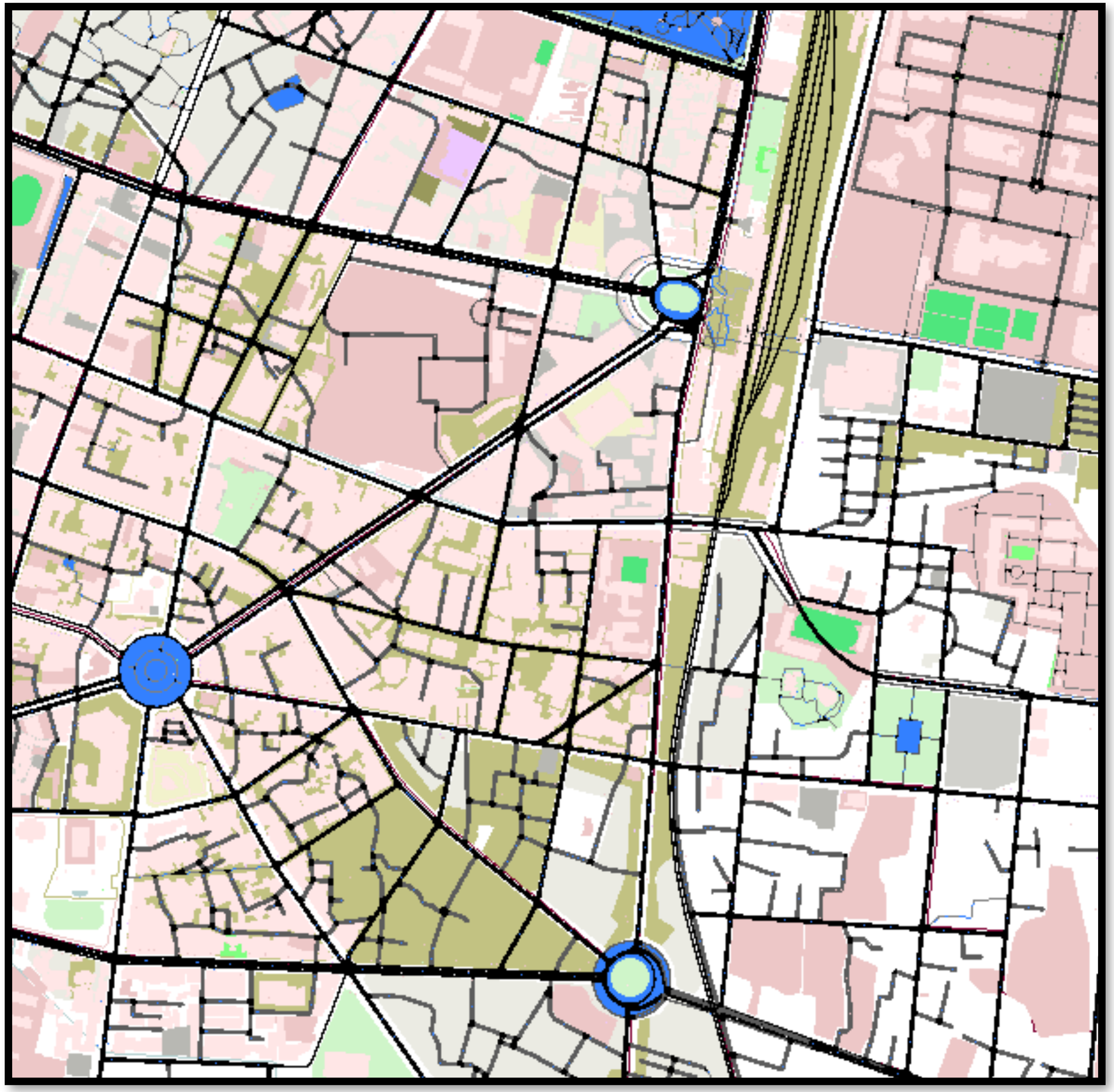
Add SumoRoad Position

- SUMO error for command 171: Position conversion requires a **source position and a position type as parameter**.

OSM



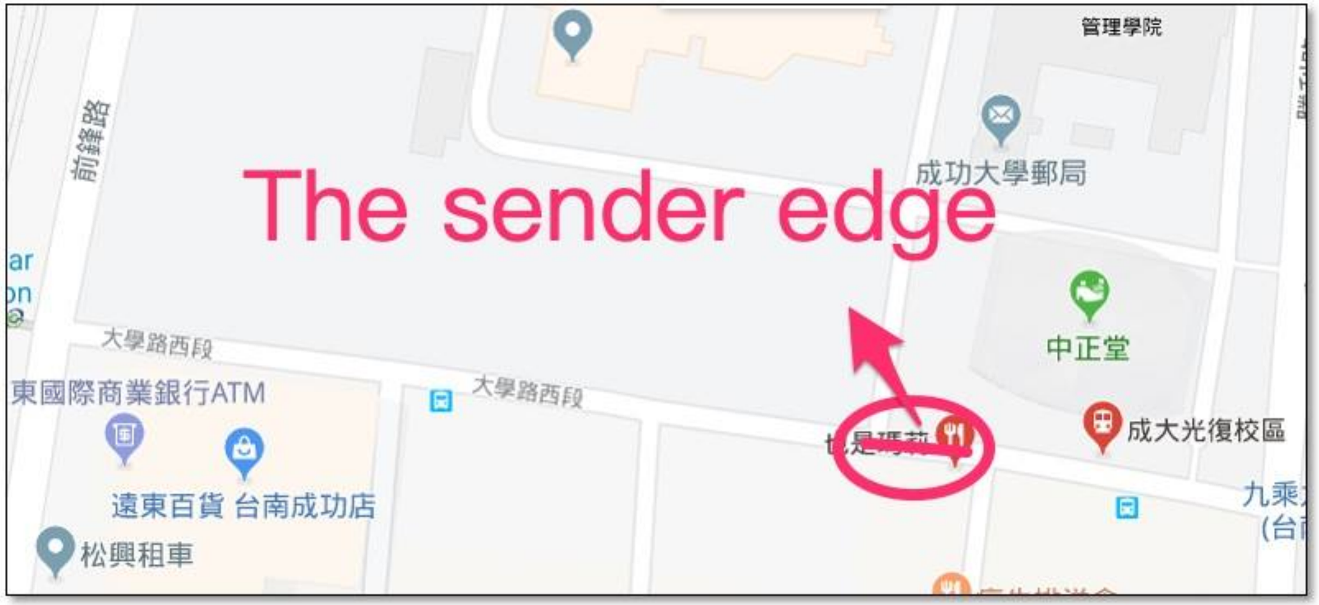
Sumo-gui

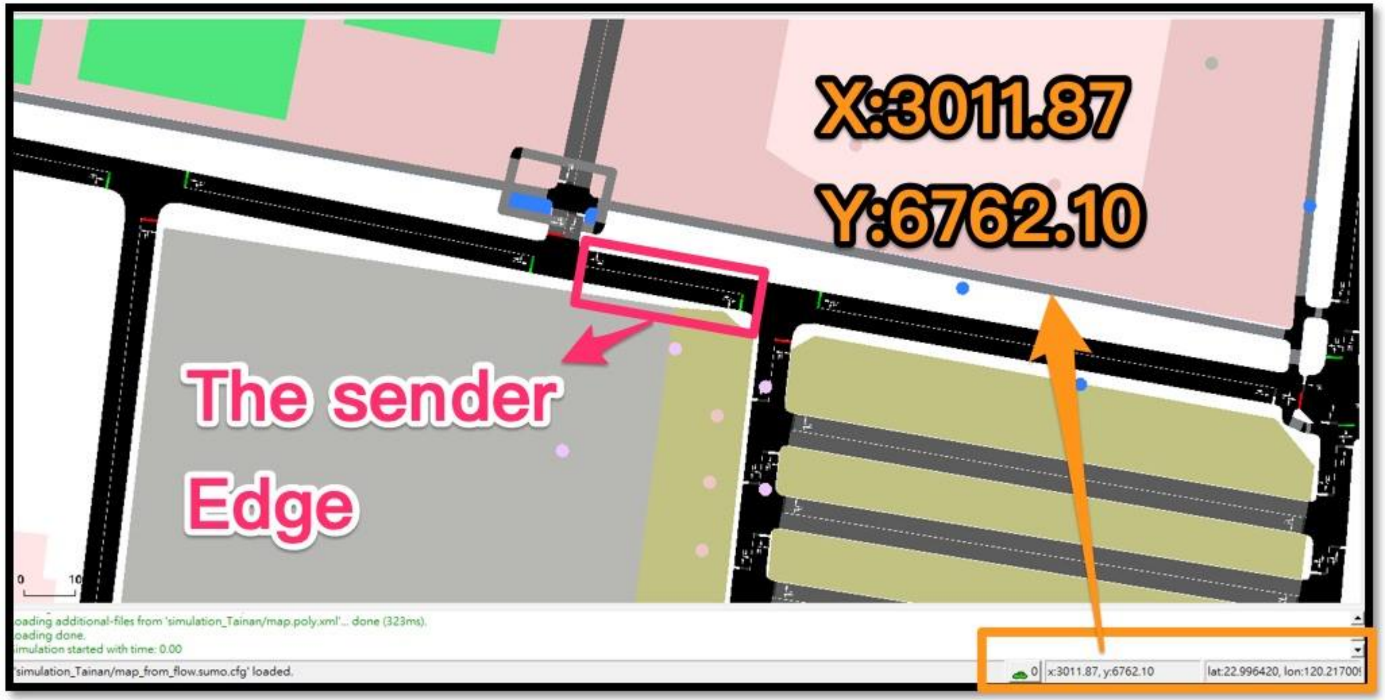


convertRoad

- ------------convert2D part-------------
- **convert2D**('496249899#1', 0.66694273848187771, (byte)0, false)
- Result:**2467.957,6810.675**

- ------------convertRoad part------------
- **convertRoad(2467.957**, **6810.675** , false, "ignoring")
- Result:496249899#1, 0.6669427384818679, 0

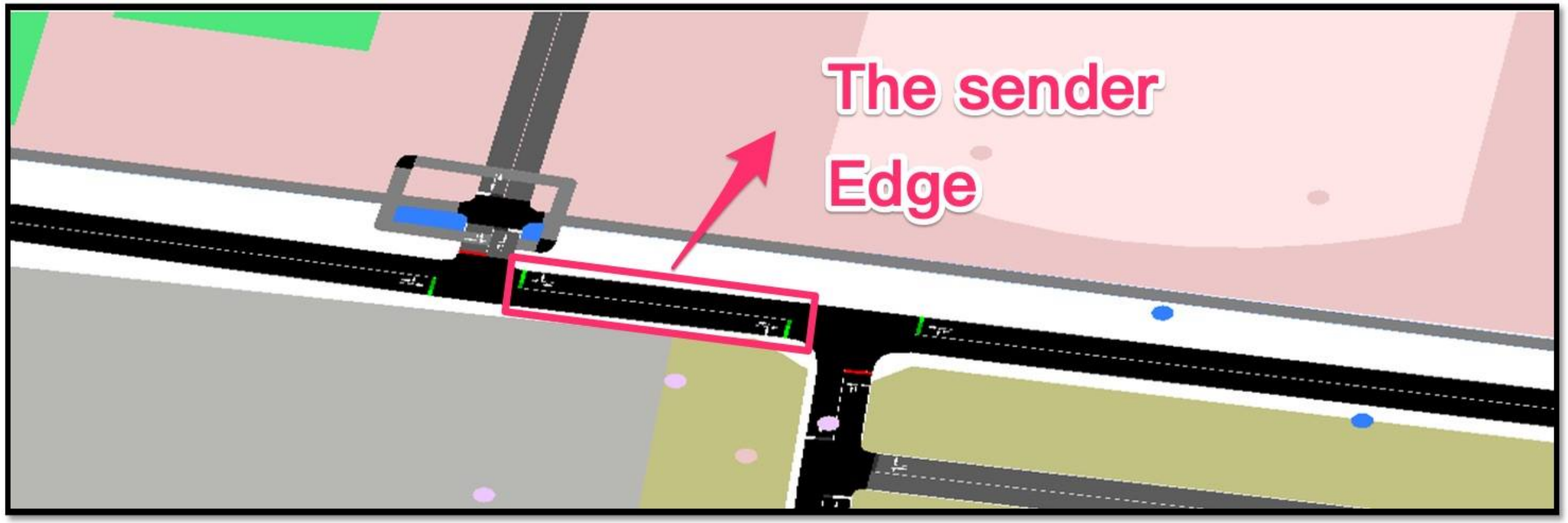




```
oading additional-files from 'simulation_Tainan/map.poly.xml'... done (323ms).
oading done.
imulation started with time: 0.00
'simulation_Tainan/map_from_flow.sumo.cfg' loaded.
```

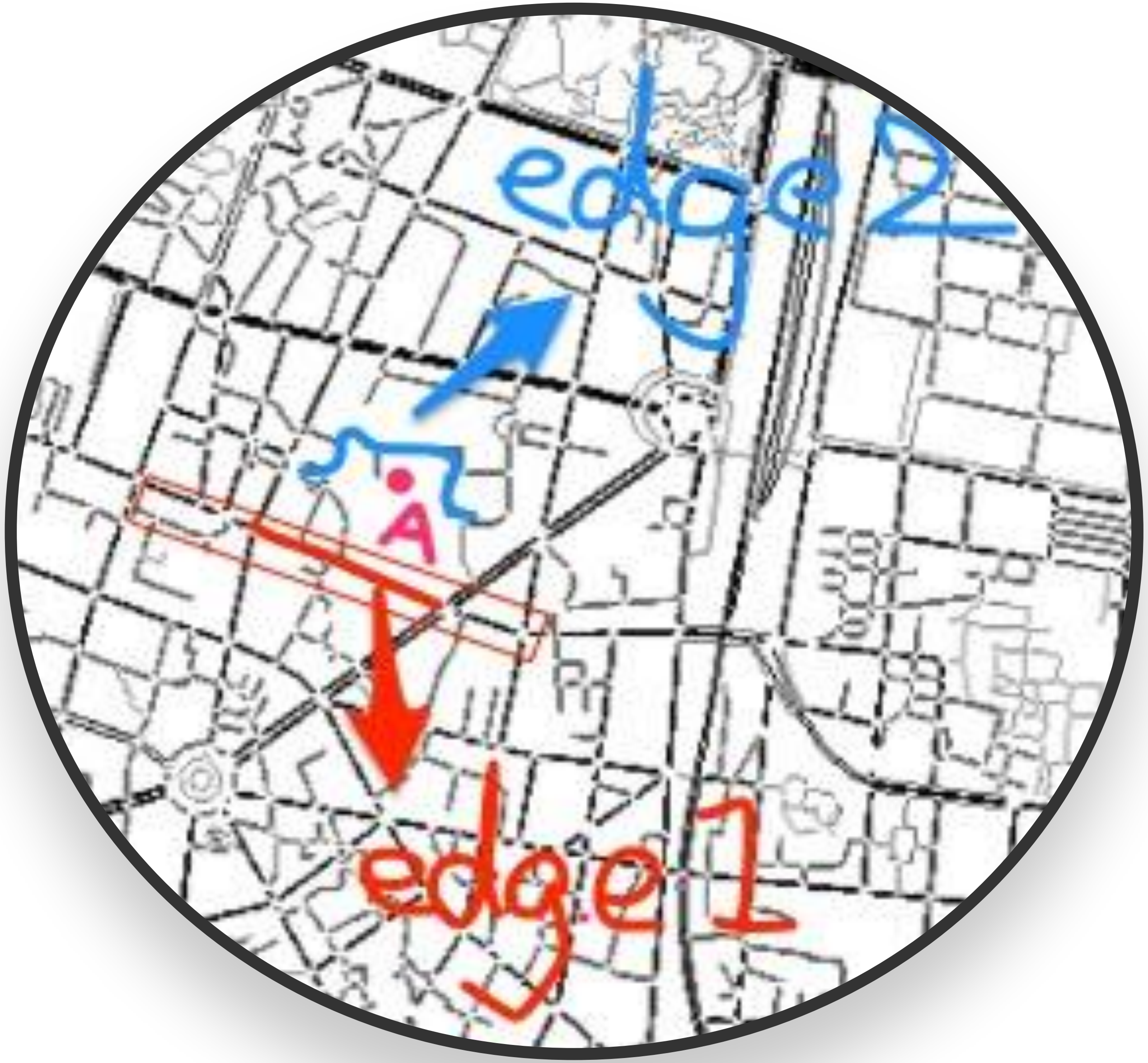
```
0 | x:3011.87, y:6762.10 | lat:22.996420, lon:120.21700!
```

- Sender address
- (X,Y)=(2989.02, 6765.41)
- (lon,lat)=(120.216786, 22.996446)
- edgeID:”-537706053#2”



```
convert2D('-537706053#2', 0.0, (byte)0, false)
3013.7,6761.51

convertRoad(2989.02, 6765.41, false, 'ignoring'))
-537706053#2, 24.98140320859787, 0
```

The **precision problem** of convertRoad



convertRoad(1387, 6497 , false, "ignoring")

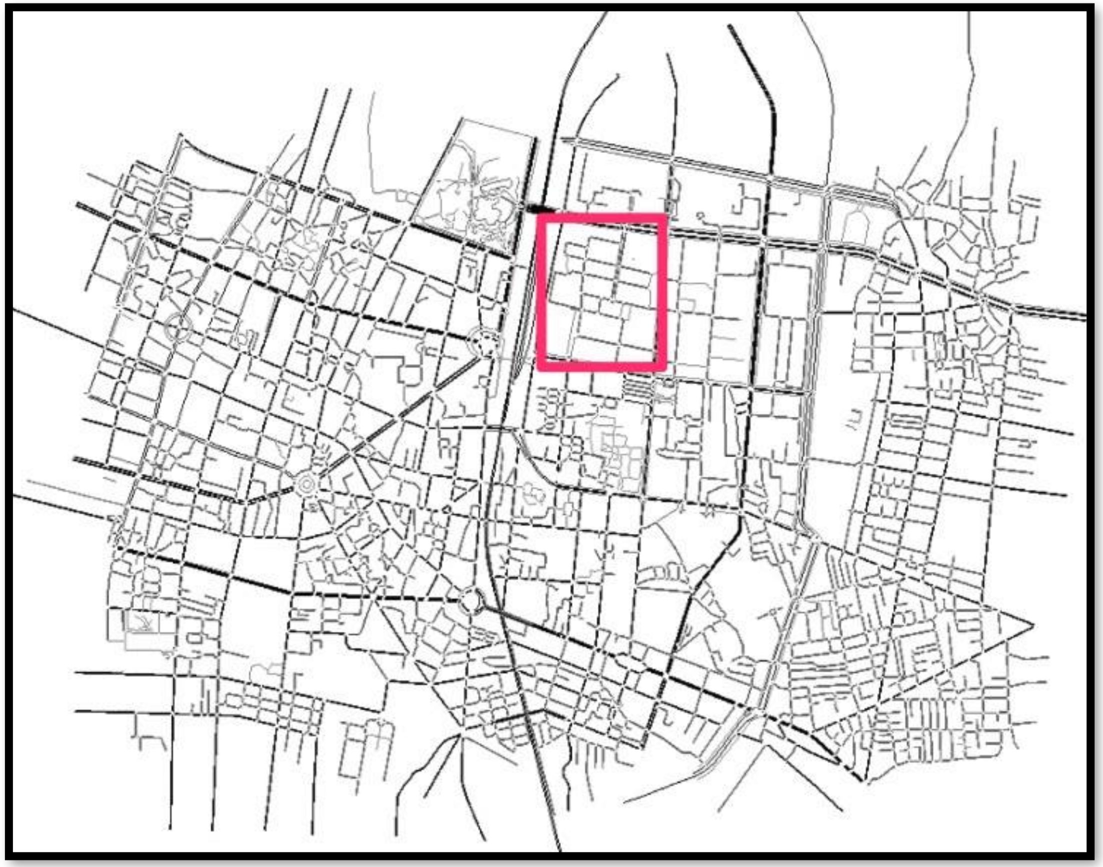
```
<edge id=":1235793433_4" function="internal">
        <lane id=":1235793433_4_0" index="0" disallow="tram rail_urban rail rail_electric ship"
speed="13.89" length="11.04" shape="1388.94,6495.41
                                        1391.56,6506.13"/>
</edge>
```

```
<edge id=":1235793433_5" function="internal">
        <lane id=":1235793433_5_0" index="0" disallow="tram rail_urban rail rail_electric ship"
speed="3.65" length="2.34" shape="1388.94,6495.41
                                        1388.44,6496.76
                                        1387.76,6497.34"/>
</edge>
```

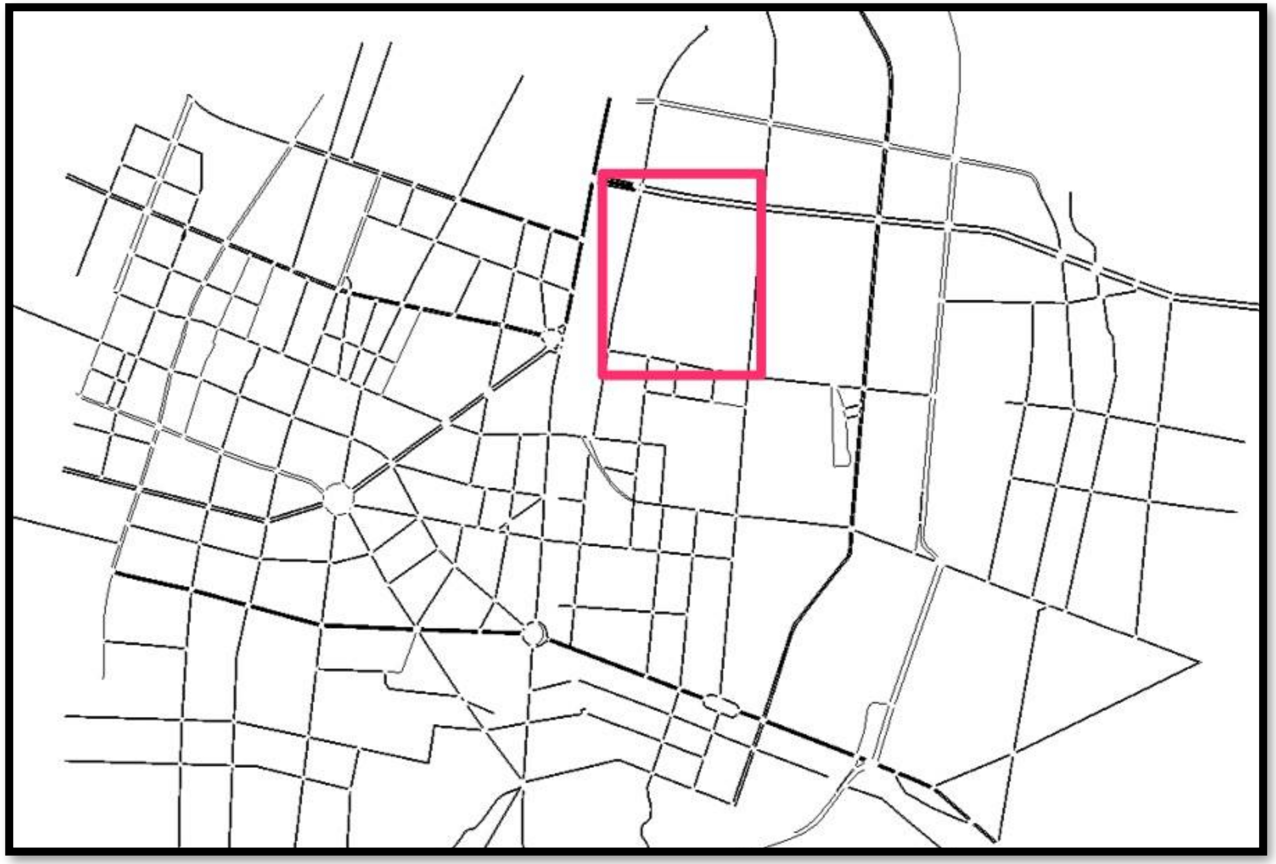
```
<edge id=":1235793433_11" function="internal">
        <lane id=":1235793433_11_0" index="0" disallow="tram rail_urban rail rail_electric ship"
speed="3.65" length="2.34" shape="1387.76,6497.34
                                        1386.89,6497.13
                                        1385.83,6496.15"/>
</edge>
```

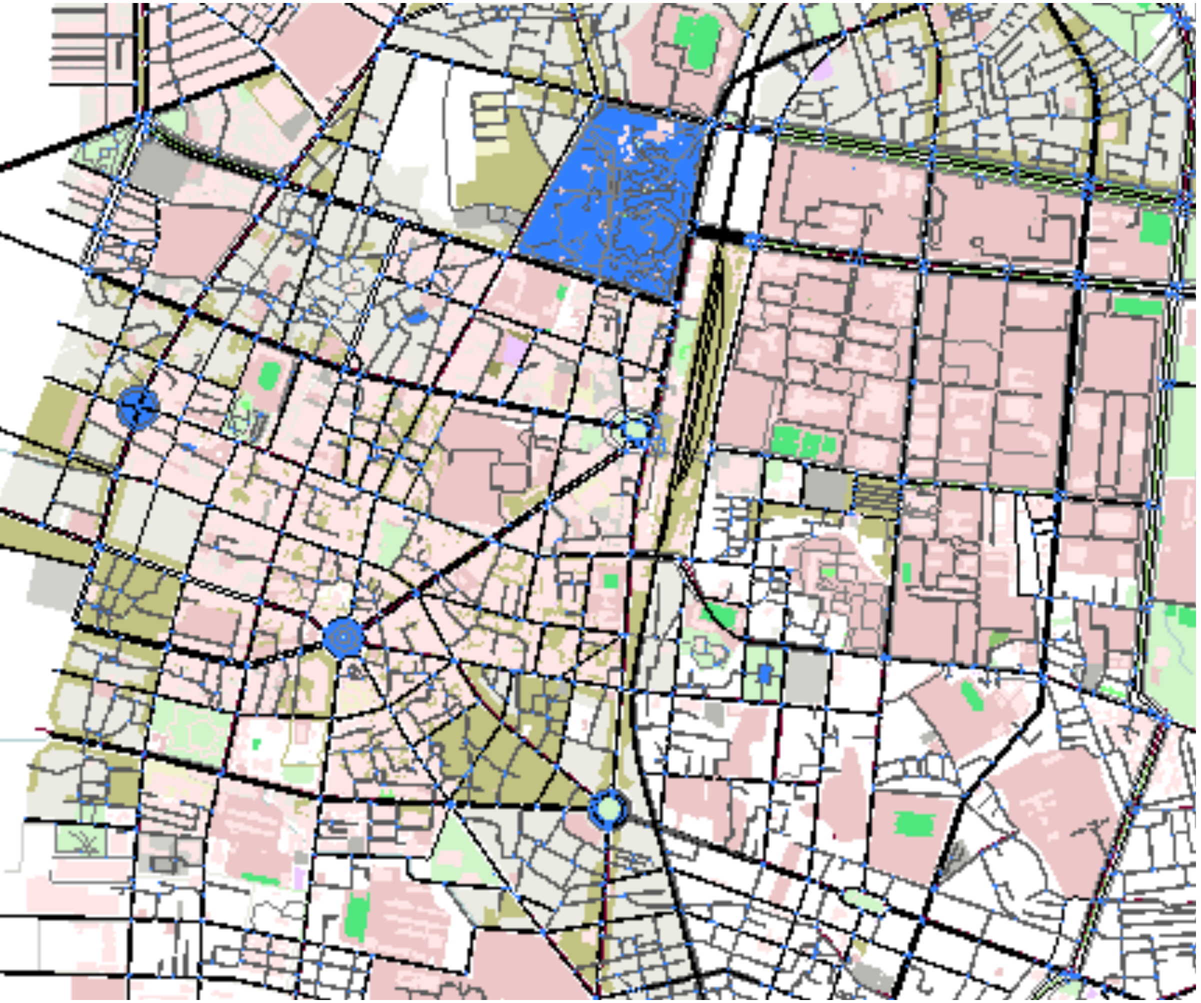
A solution to the **pinpoint problem** of convertRoad

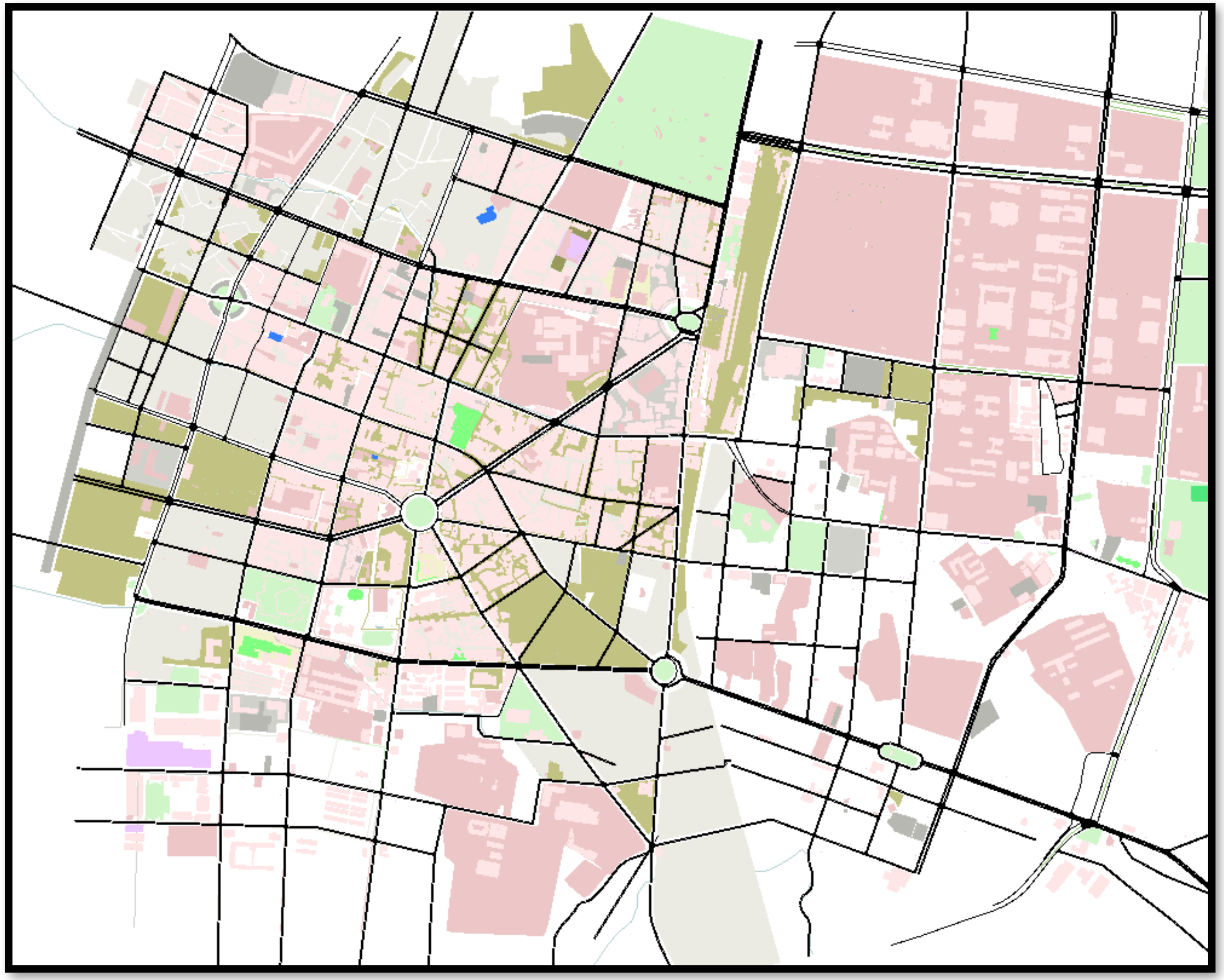
Before



The edited version







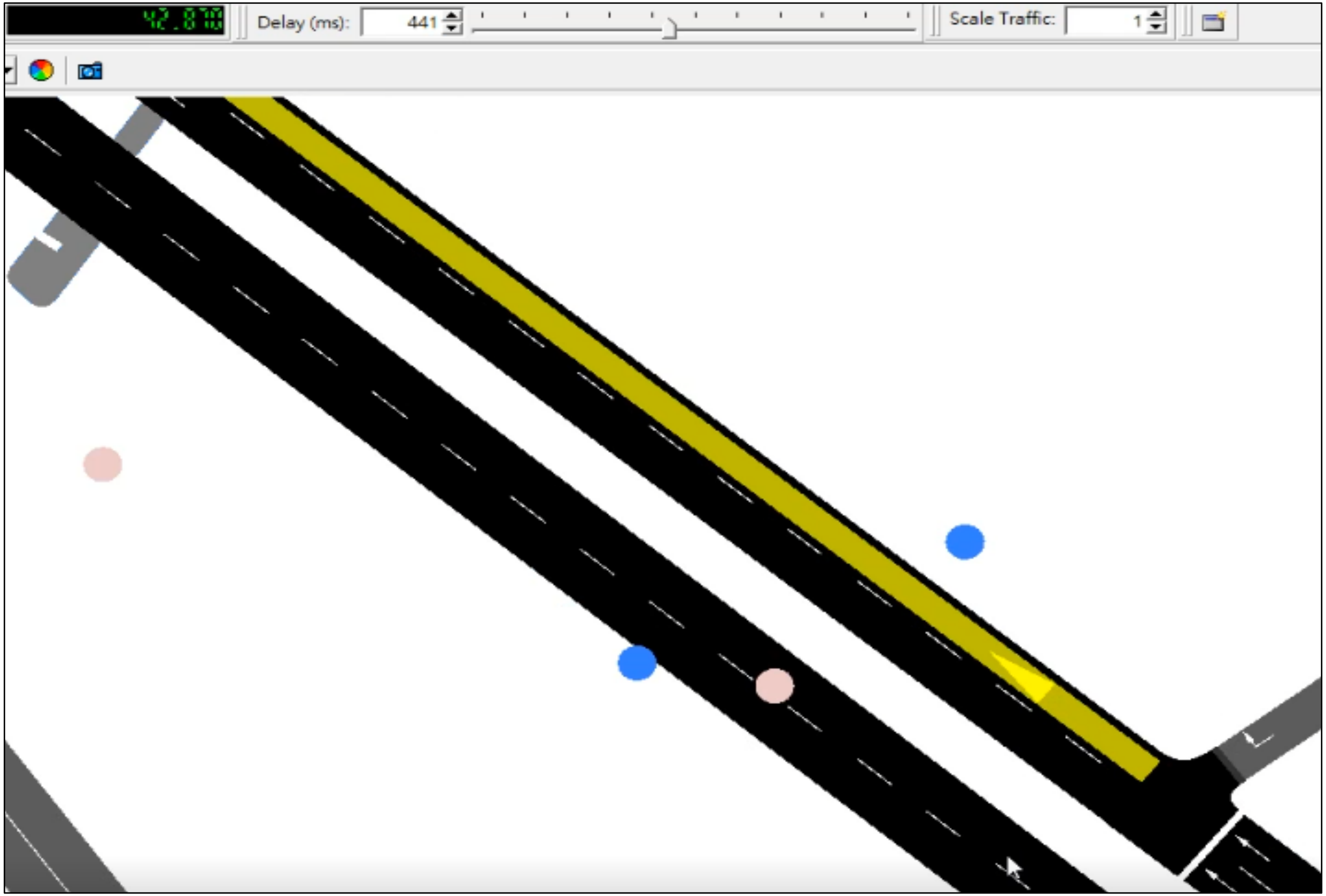
Remove the trivial lanes
and make the map simple

findRoute

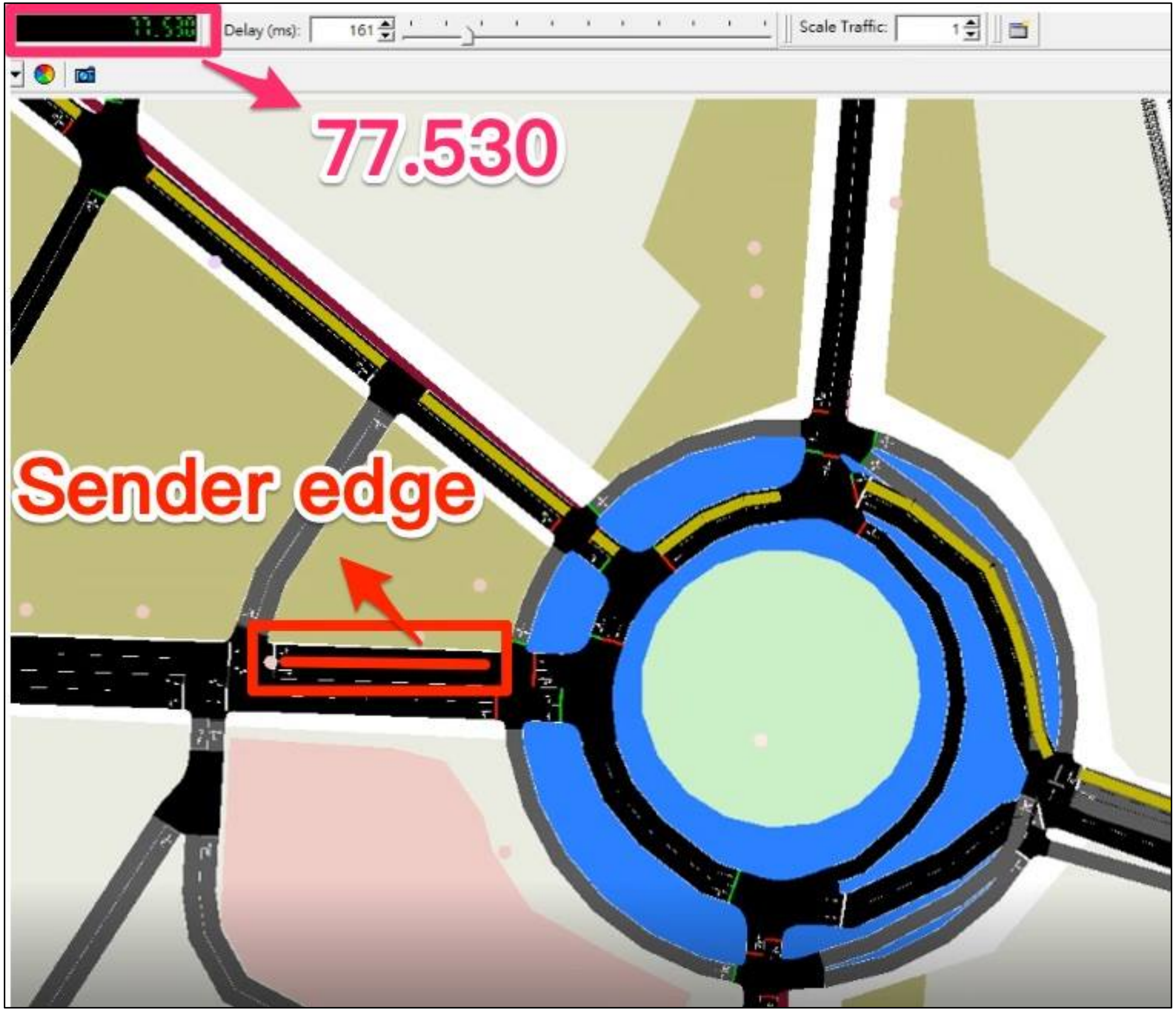
```
SumoStage stage = (SumoStage) conn.do_job_get(Simulation.findRoute(fromEdge, toEdge, vType, depart,
routingMode));
```

```
String fromEdge =
"307244665#4";
String toEdge =
"31794904#2";
String vType =
"routeByDistance";
double depart = 60.0;
int routingMode = 0;
```

```
newRoute:
[307244665#4, -298597678#5, -298597678#4, -298597678#3,
-298597678#2, -298597678#1, -298597678#0, -298597679#4,
-298597679#3, -298597679#2, -298597679#1, -298597679#0,
-413131917, -186109367#5, -186109367#4, -186109367#3,
-186109367#2, -186109367#1, -186109367#0, -316267745#1,
-316267745#0, 667223560#0, 667223560#1, 41389171,
-496261701#2, -496261701#1, -496261701#0, -41389179,
413128683, 413128684, 41389178, 160253728, 107538178#0,
31794904#0, 31794904#1, 31794904#2]
```



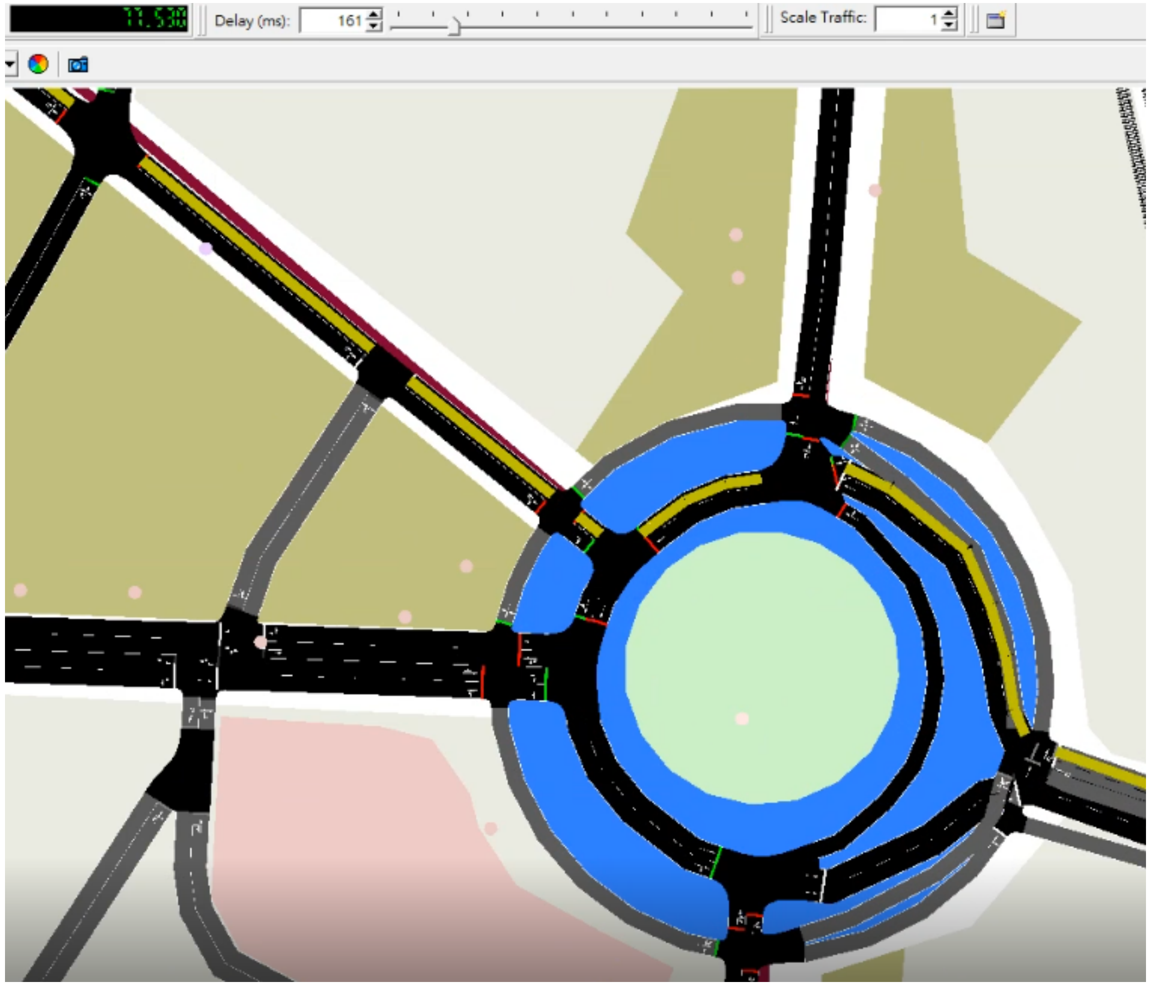
Vehicle "flow0.0" starts at 40s



Follow default route from 40s-90s and re-built the new route when the sender have the request at 90s

```
If(time==90.0){
String senderEdgeID
="160253722#1";

Vehicle.changeTarget("flo
w0.0", senderEdgeID);
}
```

77.530s

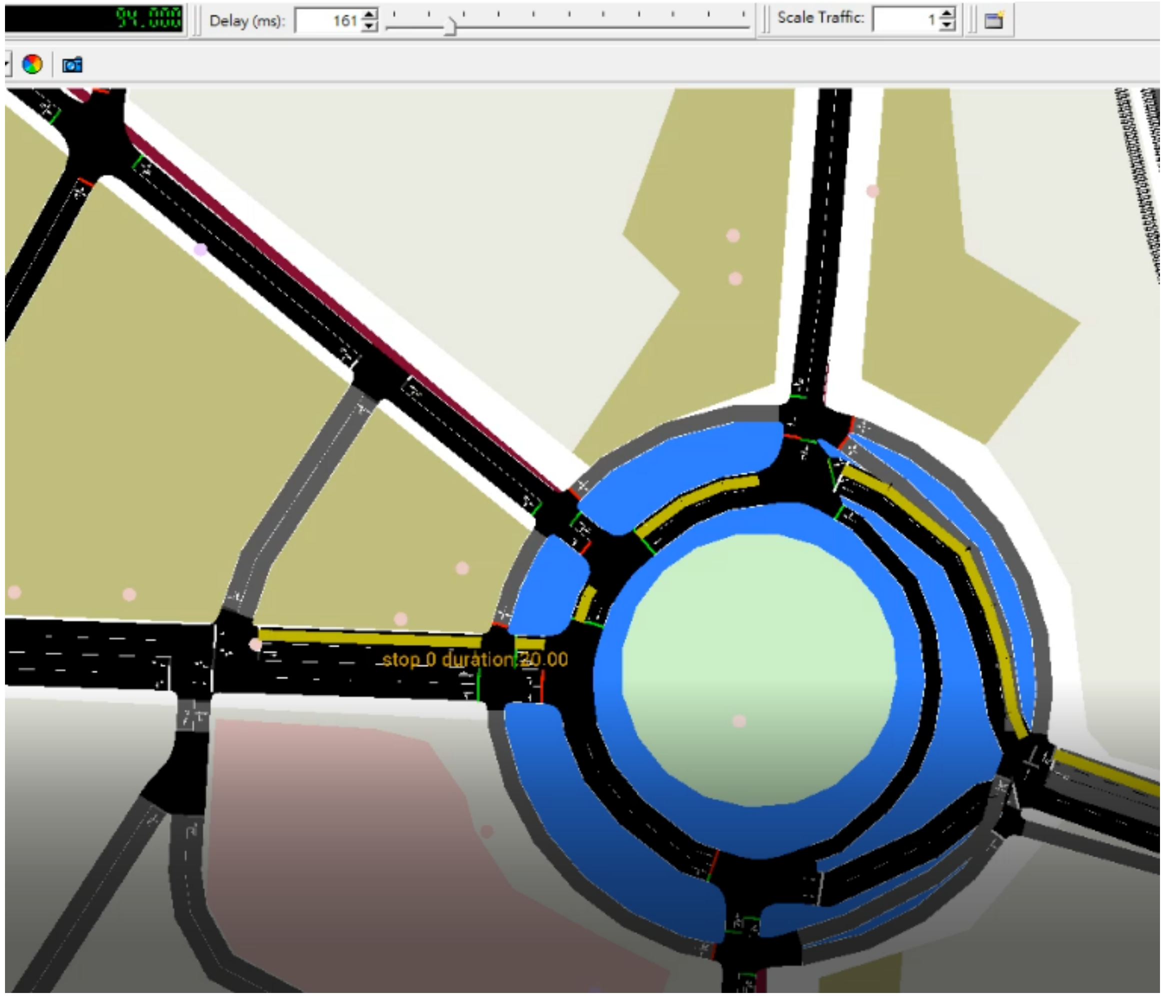


77.530

Delay (ms): 161

Scale Traffic: 1

94.000s



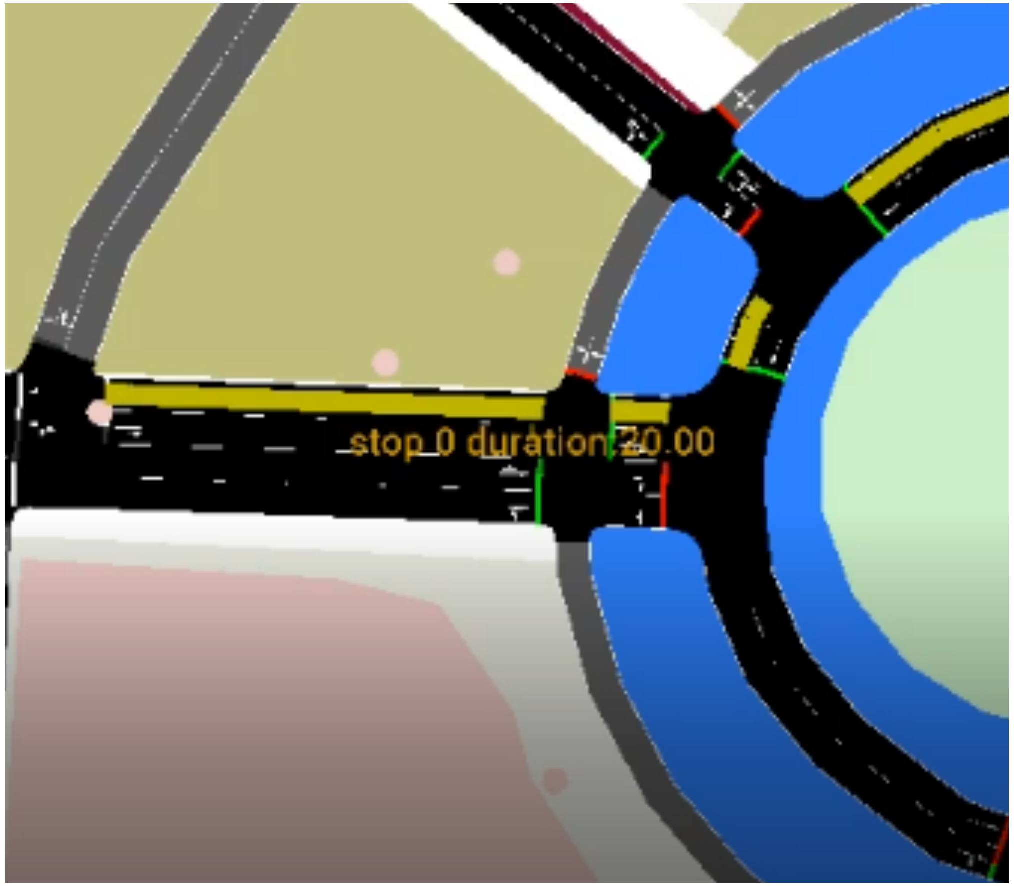
94.000

Delay (ms): 161

Scale Traffic: 1

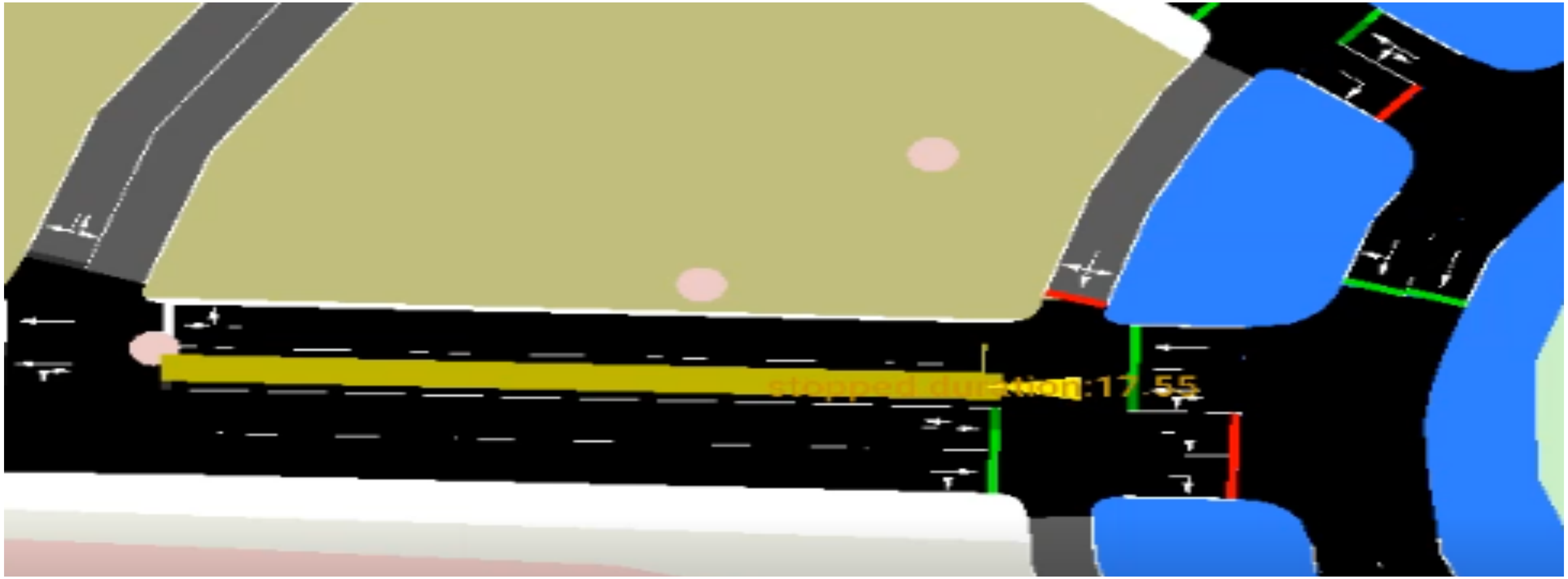
```
(Vehicle.changeTarget("flow0.0", senderEdgeID));
```

94.00s



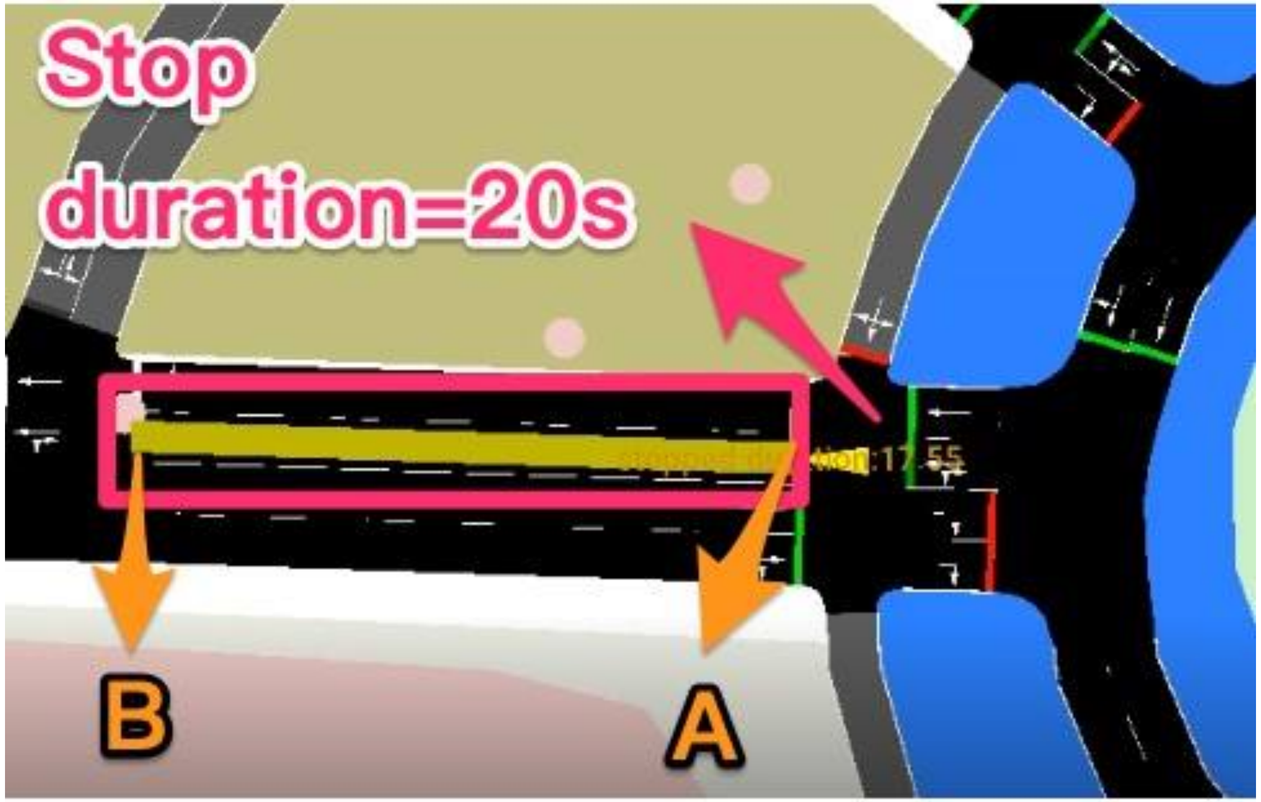
```
 if(timeSeconds==90.0) {
String senderEdgeID ="160253722#1";

changeTarget("flow0.0", senderEdgeID));

new SumoStopFlags(false, false, false, false, false);

double duration = 20.0;

setStop("flow0.0", senderEdgeID, 1.0, (byte)1,
duration, sf_send));

// 計算到達receiver的
travelTime
// 通知sender
}
```

Vehicle stopped at position 1.0m of the lane in 192.0s



Vehicle.setStop
("flow0.0", senderEdgeID, **1.0**, (byte)1,
Duration=20, SumoStopFlags));

A route after the stop of the sender



A:192s

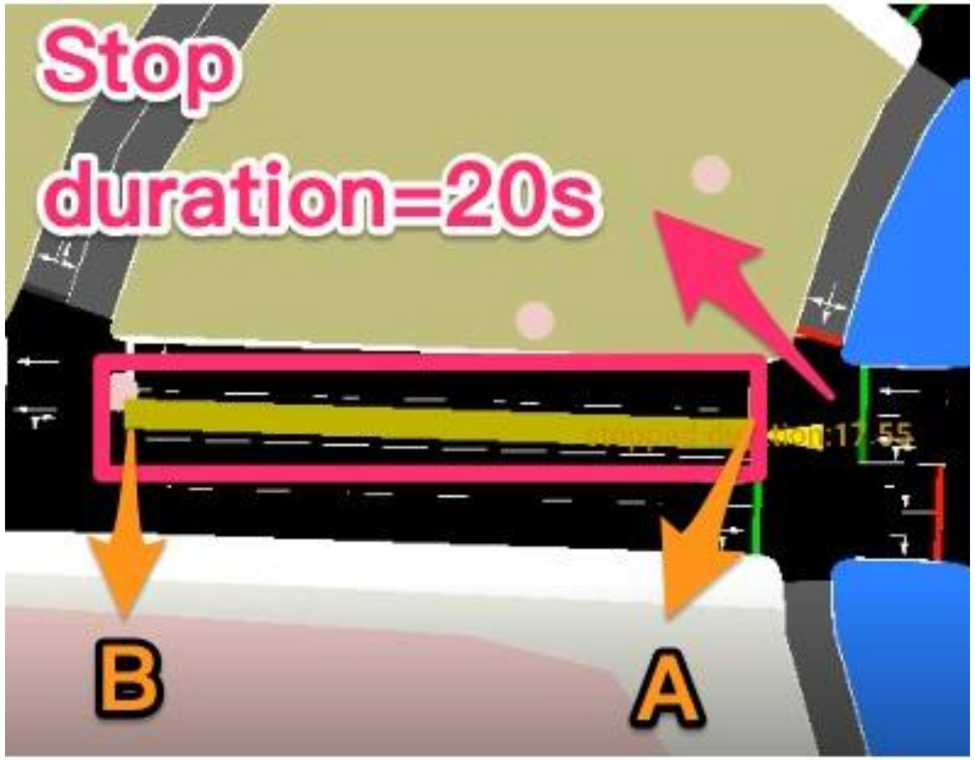
A_stop:192-212s

B:217s

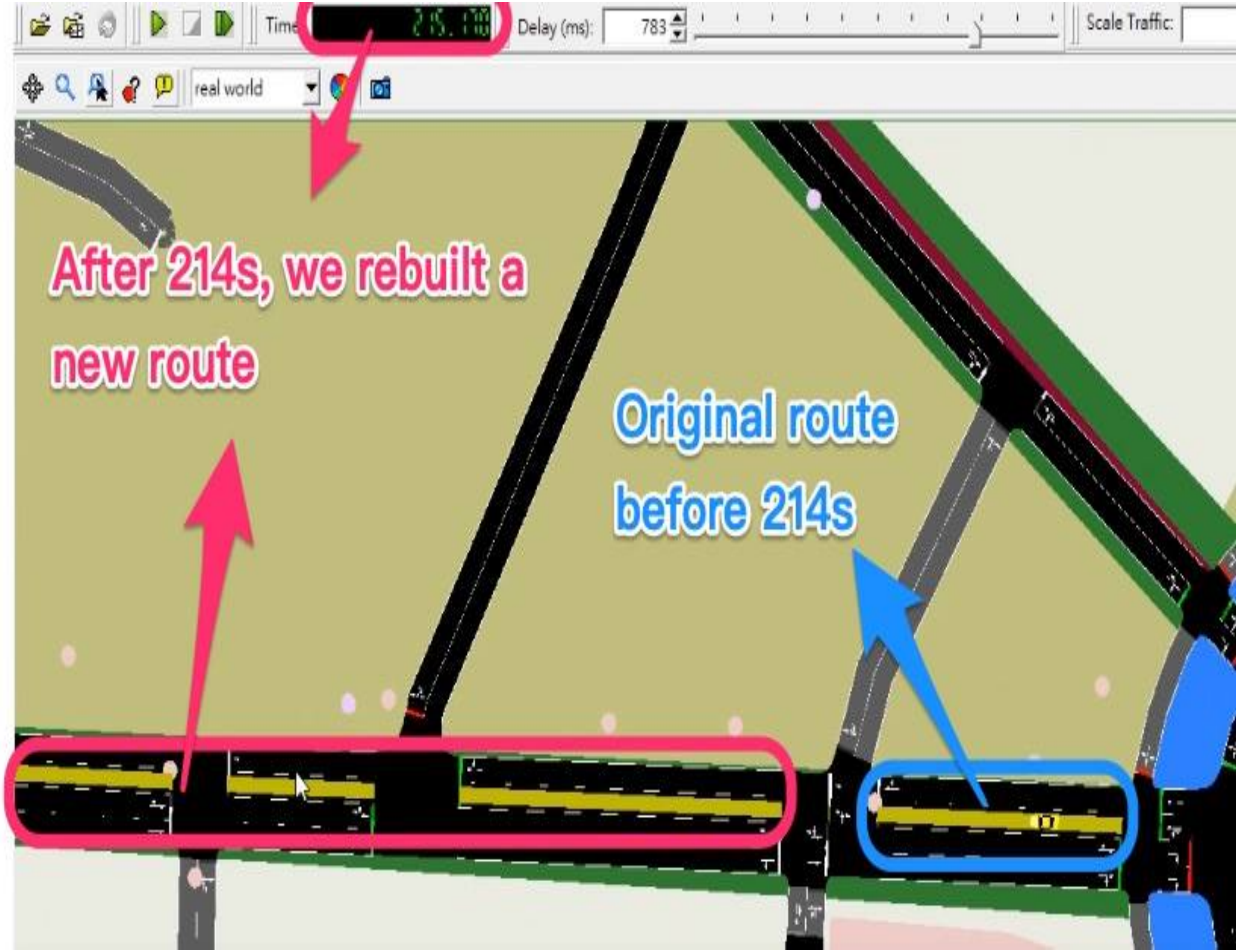
If we do not arrange the new route of this vehicle before arriving B, this car would disappear at B.

Re-build the new route

A_stop:192-212s
B:217s



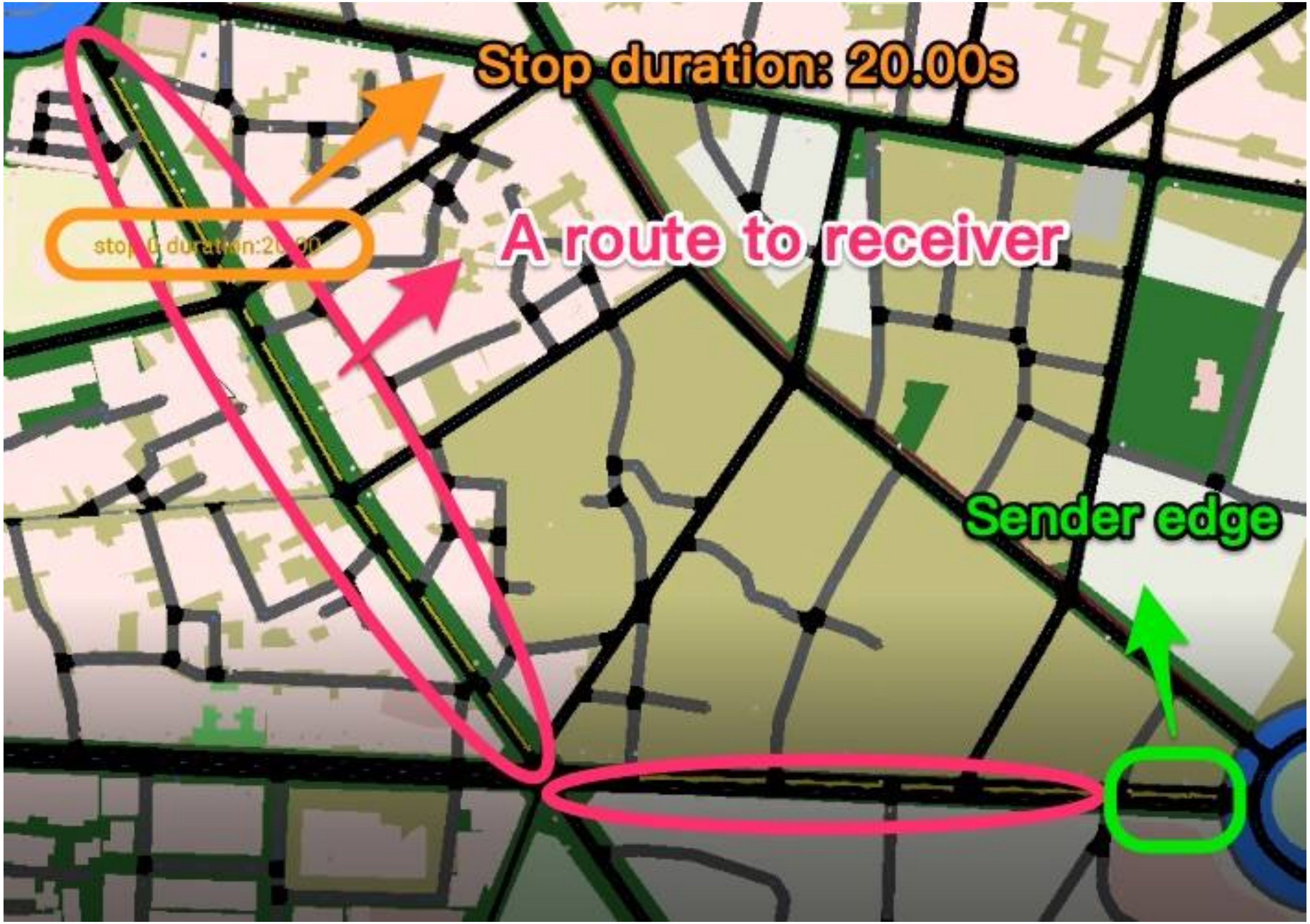
```
if(timeSeconds==214.0) {
String receiverEdgeID = "-
279032146#1";.
changeTarget("flow0.0",
receiverEdgeID));

new SumoStopFlags
double duration = 20.0;

setStop("flow0.0", receiverEdgeID, 1.0,
(byte)0, duration, sf_rec));

}
```



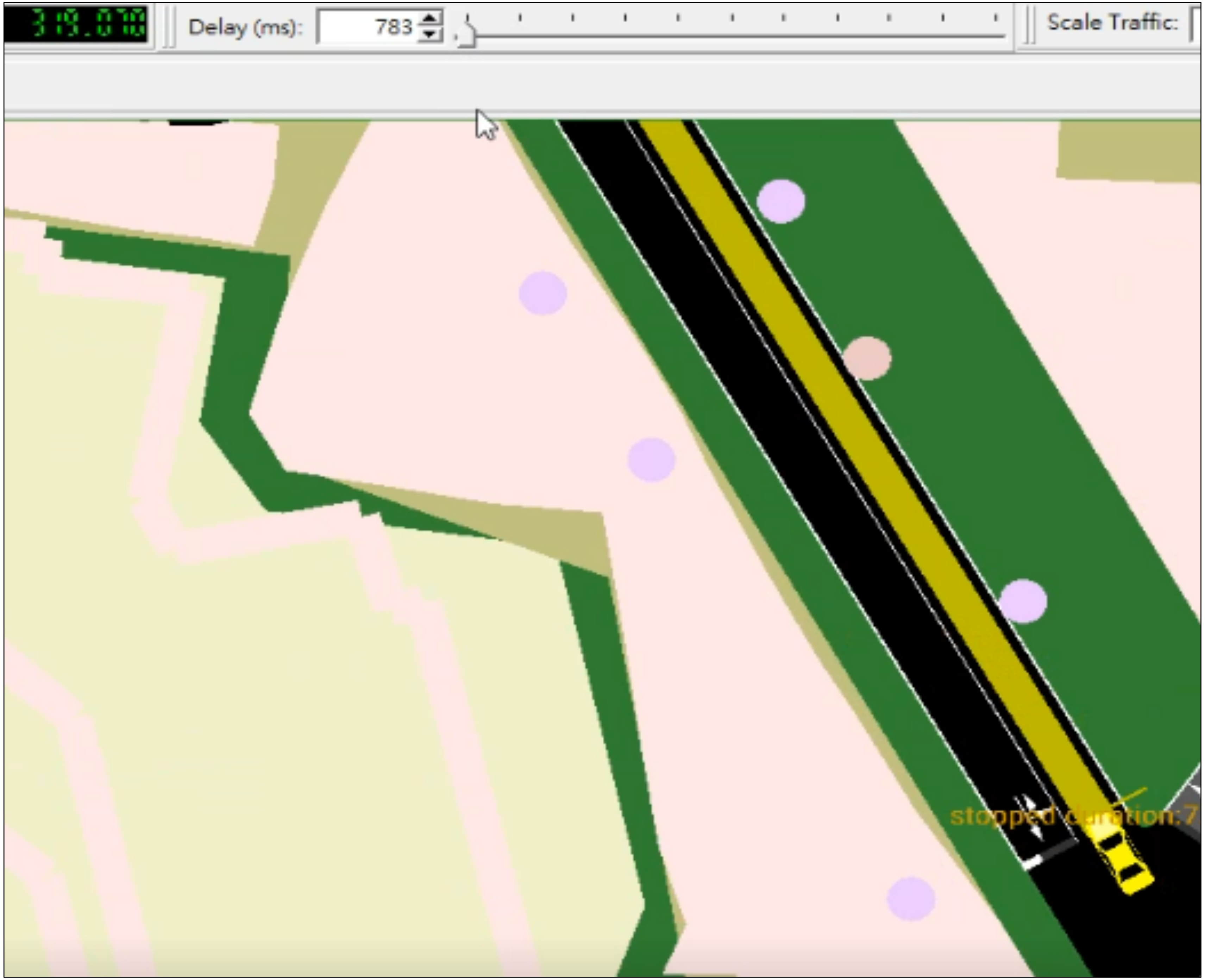
After 214s, we rebuilt a new route

Original route before 214s

```
if(timeSeconds==214.0)
{
String receiverEdgeID
= "-279032146#1";.

changeTarget("flow0.0"
, receiverEdgeID));

}
```


Stop duration: 20.00s
stop 0 duration:20.00
A route to receiver
Sender edge

215.17s



The car should stop for 20s in the edge of the receiver.

Next step (04/29-05/05)

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)