

Update at the end of the file is the recent changes.

-----  
**Files added** to incorporate External mobility file are :

- >mobility\_parser.c
- >mobility\_parser.h
- >hashtable.c
- >hashtable.h
- >xmob.c
- >xmob.h

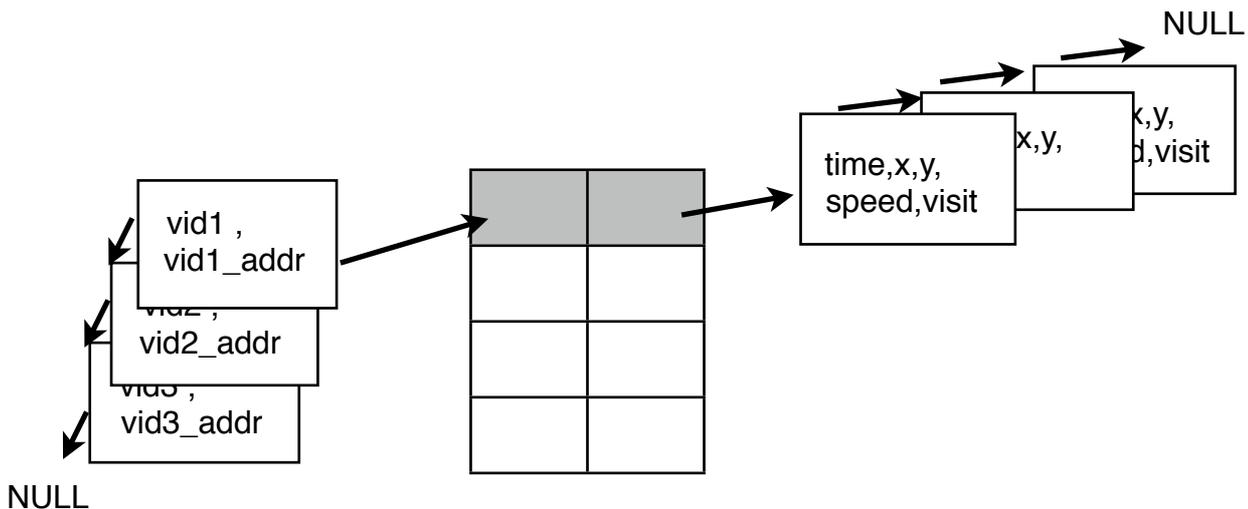
File : mobility.txt

and the **changes to the existing files** are :

- openair2/LAYER2/Makefile.inc
- openair2/UTIL/OMG/common.c
- openair2/UTIL/OMG/defs.h
- openair2/UTIL/OMG/omg.c
- openair2/UTIL/OMG/omg.h
- openair2/UTIL/OMG/omg\_constants.h

-----  
Mobility description of nodes read from file are stored using the following structure :

Here hash table is used to easy access the mobility description when ever needed with less complexity. Even when out of order description is given , sorting is done in the 2nd linked list (right most in the pic)



Linked list holding vehicle id and a address pointing to an entry in hash table which further points to a linked list holding all the mobility description to the vehicle id .

The mobility file mobility.txt has the following format :

**syntax** : time          node\_id          x          y          speed

good mobility description : in order w.r.t time

e.g:

```
0 11 1 1 0.0
0 22 1 2 0.0
0 33 1 3 0.0
0 44 1 4 0.0
5 11 5 5 2.0
5 22 5 6 2.0
5 33 5 7 2.0
5 44 5 7 2.0
```

other: out of order w.r.t

e.g:

```
0 44 1 4 0.0
5 11 5 5 2.0
5 22 5 6 2.0
5 33 5 7 2.0
0 33 1 3 0.0
5 44 5 7 2.0
0 11 1 1 0.0
0 22 1 2 0.0
```

disconnected: vehicles may end their trips earlier and have to sleep before they have the next mobility description.

e.g:

```
0 11 1 1 0.0
5 11 5 5 2.0
15 11 10 10 12.0
```

mobility\_parser and hashtable :

### **Main functions :**

read\_mobility\_file() reads the mobility file and generates a hash table and linked list , later on all functions use this table to fetch required data.

Two main functions used in xmob.c are [eXternal MOBility]

get\_next\_position (table,vehicle\_id):

each time this function is called returns the next target to move and marks it with visit=1, function when called again returns the target with visit=0 and marks it as 1. (if description is given out of order a sorting is already done right after the mobility file is read)

reset\_visit\_status(table,time,vehicle\_id) :

This function is used because of the behavior of the `get_next_position` and for mobility behavior where node travels to a particular target and its next description is after a long time, during this period the node is put to sleep till the next movement mentioned in the mobility file. in order to decide whether to sleep or move the next description is fetched, since `get_next_position` already marks `visit = 1`, we need to correct it.

## UPDATE :

--> In the code RWP and other, the `cur_time` is initialized to 1.0, for that reason the node mobility description need to start from time 1, meaning that at 1 initial position are mentioned and movement is described from time step 2

```
1 id initial_x initial_y 0
2 id x_target y_target 0
so..on
```

--> In function `update_xmob_nodes()`:

once the node finishes its mobility description, two options are added either to put the node to long sleep with period 999 (disabled) or just to return NULL so that the job for the node is not added to `Job_Vector` (enabled).

--> Need to add interface for getting file from command line, option is provided in `omg_param_list`.

--> Input on simulation time was needed, (not clear with this point)

--> need to scale for larger simulation (by completely removing the trace of the node from hashtable, API is made available in hashtable but need to invoke. and also from the linked list if number of nodes is large, time concern).

--> All nodes are created at the start, need to look up and create with time (not sure)

--> makefile in the folder LAYER2 is modified to access the executables generated in OMG folder, modified version is in `LAYER2_Makefile` in shared folder.

asn1c version 0.9.22 is the one needed for OPENAIR which cannot be directly available from site or from ubuntu apt-get, it is only available from svn checkout.