Instruction of the signal timing design in DTALite

--Phase based signal representation model

Provided by Jun Zhao (jzhao124@umd.edu),
Xuesong Zhou (xzhou74@asu.edu) and Lei Zhang(lei@umd.edu)

The training lecture on signalized intersections modeling can be viewed on the YouTube: (https://www.youtube.com/watch?v=vyFfAVSeDYg)

This document aims to help users to understand input settings for signal scenarios implemented in DTALite which consists of the following seven sections. All the datasets introduced in this documentation could be found in the **Phase based signal representation model – testing dataset**.

1. Section 1 shows the data requirement and setting process for performing signal scenario in DTALite;
2. Section 2 presents a sample case and provides the guidelines about data preparation and parameter specification in input files;
3. Section 3 mainly introduces parameter settings in key simulation configuration files;
4. Section 4 mainly illustrates how to use the NEXTA GUI to show phase plan in each intersection, how to edit the existing phases, and add/delete phases;
5. Section 5 gives a step by step instruction of how to import data from synchro to DTALite. Here, we use a set of Synchro data from interstate 270 as an example. An eight-step method will be introduced;
6. Section 6 introduces the default signal plan setting in NEXTA;
7. Section 7 explains some key things that need to pay attention before running the simulation.

Content

There are two models to deal with the signal timing in DTALite which are listed in the following flow charts. Method 1 is **link based signal representation model** and model 2 is **phase based signal representation model**. This documentation introduces model 2.

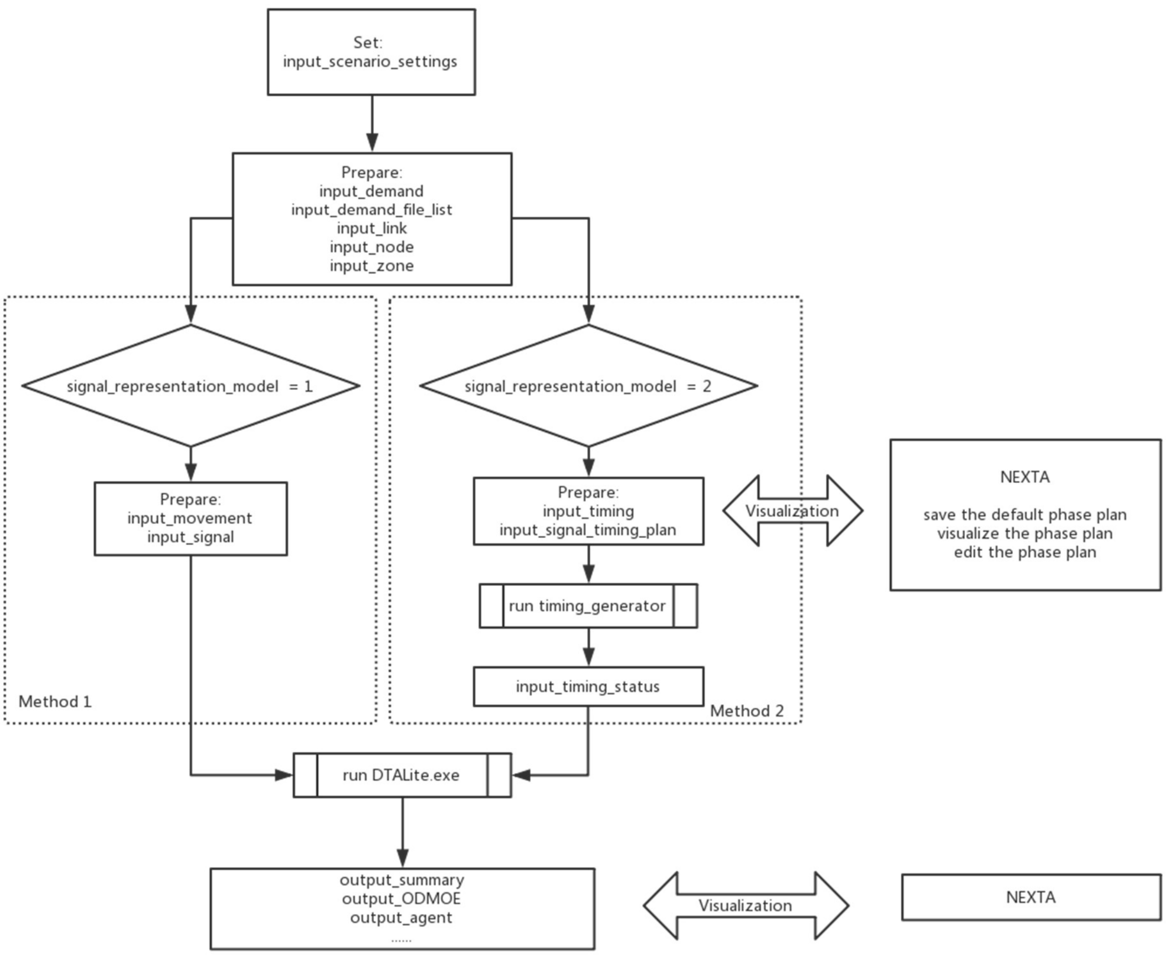


Figure 1. Flow chart of simulation with signal timing

Section 1: Key files required for signal scenario settings

The key input setting files and output files for signal scenario are shown in Table 1.

Table 1. Key files for signal scenario settings

File Group	Files list	Remark
Input files	1. input_demand.csv	Demand content for specifying distribution (observed flows of intersections)
	2. input_demand_file_list.csv	Demand files specification, loading multiplier, specify departure time distribution
	3. input_zone.csv	TAZ definition
	4. input_link.csv	Properties of all links in network
	5. input_node.csv	Properties of all nodes in network
	6.input_timing.csv*	The phase phan in each timing plan of every intersection
	7.input_signal_timing_plan.csv*	The start time, end time and other information of the different time plans of different intersections
	8. input_timing_status.csv*	The second to second phase plan of every intersection, which will be automatically generated by the **timing_generator.exe**
	9. input_scenario_settings.csv	Specify signal representation model and assignment settings. **signal_representation_model** should be set in **input_scenario_settings.**
	10. input_activity_location.csv	Relationship between zone and node
	……	
Output files	1. output_agent.csv	Statistics of agents such as path node/time sequence, OD, departure/arrival time
	2. output_summary.csv	General statistics of simulation results for each iteration
	……	

***6, 7, 8: input_timing.csv, input_signal_timing_plan.csv, input_timing_status.csv** are the new input files.

There are altogether 3 kinds of signal timing controlling methods inside the DTALite, namely:

- **signal_model_continuous_flow = 0,** the first signal model (signal_model_continuous_flow) doesn't consider signal;
- **signal_model_link_effective_green_time = 1,** the link based signal representation model, need input_signal.csv and input_movement.csv;
- **signal_model_timing_status = 2,** the Phase Based Signal Representation Model takes the **input_timing.csv**, **input_signal_timing_plan.csv** and **the input_timing_status.csv** as the input files;

Table 2 below gives an example of the parameter settings inside input_scenario_settings.csv.

Table 2. Parameter settings in file input_scenario_settings.csv

scenario_no	scenario_name	number_of_iterations	traffic_flow_model	signal_representation_model	traffic_analysis_method
1	test1	10	2	2	0

Remark:

1) the signal_representation_model of ”2” is required to enable signal scenario assignment on phase based signal representation model.

2) “number_of_iterations” defines the total numbers of iterations for scenarios corresponding to “scenario_no”.
3) the traffic_flow_model of ”2” is dedicated to spatial queue model.

Section 2: Input data preparation

A fixed-time controlled signal is chosen as a simple case in this guide. The detailed information is shown in figure 2 and figure 3. (dataset stored in **0-Signal_test_dataset_5_node_network**, and all the related input files are prepared)

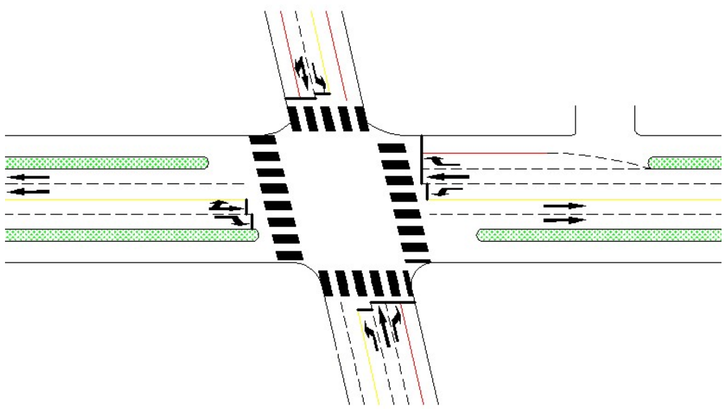


Figure 2. Geometry construction of sample intersection

In this case, the traffic network including 5 nodes and 8 links are built in NEXTA, as shown in Figure 3.

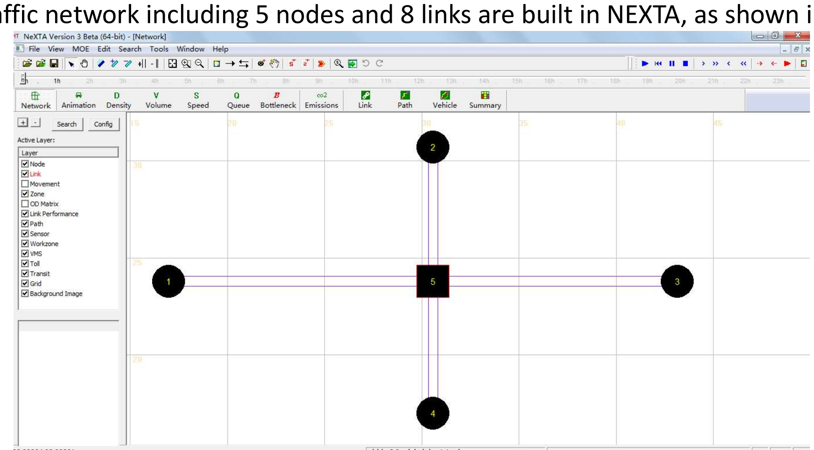


Figure 3. Traffic network displayed in NeXTA

As described above, the required basic input data files for signal scenario include the demand data (input_demand.csv and input_demand_file_list.csv) and the traffic network data (input_node.csv, input_link.csv, input_zone.csv). The detailed format of input files based on sample data are listed in **Table 3-7**.

Table 3. Sample data in file input_demand.csv

from_zone_id	to_zone_id	number_of_trips_demand_type1
1	2	100
1	3	100
1	4	100
2	3	100
2	4	100
2	1	100
3	4	100
3	1	100
3	2	100
4	1	100
4	2	100
4	3	100

Table 4. Sample data in file input_activity_location.csv

zone_id	node_id	external_OD_flag
1	1	0
2	2	0
3	3	0
4	4	0
5	5	0
…	…	…

Table 5. Sample data in file input_zone.csv

zone_id	production	attraction	geometry
1	0	0	<Polygon><outerBoundaryIs><LinearRing><coordinates></coordinates></LinearRing></outerBoundaryIs></Polygon>
2	0	0	<Polygon><outerBoundaryIs><LinearRing><coordinates></coordinates></LinearRing></outerBoundaryIs></Polygon>
3	0	0	<Polygon><outerBoundaryIs><LinearRing><coordinates></coordinates></LinearRing></outerBoundaryIs></Polygon>
4	0	0	<Polygon><outerBoundaryIs><LinearRing><coordinates></coordinates></LinearRing></outerBoundaryIs></Polygon>

Table 6. Sample data in file input_link.csv

from_node_id	to_node_id	length	number_of_lanes	number_of_leftturn_lanes	speed_limit	jam_density
1	5	0.625	2	**1**	25	200
5	3	0.625	2		25	200
2	5	0.625	2	**1**	25	200
5	2	0.625	1		25	200
5	4	0.625	1		25	200
4	5	0.625	3	**1**	25	200

3	5	0.625	3	**1**	25	200
5	1	0.625	2		25	200

Remark:
1) Number of lanes: the number of lanes on specific links.
2) The value of "Number of left-turn lanes" should be specified when left-turn lanes exist (**including separated left-turn lanes and shared left-turn lanes)**

Table 7. Sample data in file input_node.csv

node_id	control _type	control_type _name	x	y	geometry
1	0	unknown_control	16.97894	23.79017	<Point><coordinates>16.978937,23.790170</coordinates></Point>
5	**5**	pretimed_signal	30.58159	23.79017	<Point><coordinates>30.581594,23.790170</coordinates></Point>
3	0	unknown_control	43.16913	23.79017	<Point><coordinates>43.169128,23.790170</coordinates></Point>
2	0	unknown_control	30.58159	30.69301	<Point><coordinates>30.581594,30.693011</coordinates></Point>
4	0	unknown_control	30.58159	16.98884	<Point><coordinates>30.581594,16.988842</coordinates></Point>

Remark:
1) The control type of intersection nodes should be specified as "pretimed signal" (as shown node 5 in the table above). 0 here means unknown control.
2) For the detailed node type information, please check **input_node_control_type.csv**.

Section 3: Parameter settings in key simulation configuration files

First, as mentioned in table 2, set **signal_representation_model = 2** in input_scenario_settings.csv.

Then, prepare **the input_signal_timing_plan.csv** and **input_timing.csv** tables.

The input_signal_timing_plan.csv file looks like this:

Table 8. Sample data in file input_signal_timing_plan.csv

[TOD]	int_id	timing_plan_no	start_time_in_sec	end_time_in_sec	starting_phase	off_set
	5	1	0	86400	1	11

int_id: intersection node id;

timing_plan_no: the timing plan of this intersection, for example, we could have different phase plans in one intersection during peak hour and off-peak hour (could define maximum 20 different timing plans in one day). It should have the same number of timing plans for one intersection with the file of input_timing.csv (will be introduced next);

start_time_in_sec: the start time of the timing plan. (unit in seconds);

end_time_in_sec: the end time of the timing plan. (unit in seconds);

starting_phase: the starting phase of this timing plan;

off_set: the time relationship between coordinated intersections.

The **input_timing.csv** looks like this:

Table 9. Sample data in file input_timing.csv

[Signal]	int_id	timing_plan_no	phase_id	next_phase_id	green_duration	movement_str	movement_dir_str
	5	1	1	2	120	1_3_T; 1_2_L; 3_1_T; 3_4_L	WBT; WBL; EBT; EBL;
	5	1	2	1	70	2_4_T; 2_3_L; 4_2_T; 4_1_L	SBT; SBL; NBT; NBL;

int_id: intersection node id;

timing_plan_no: the timing plan of this intersection, for example, we could have different phase plans in an intersection during peak hour and off-peak hour (could define maximum 20 different timing plans in one day);

phase_id: the phase id of this phase. We could define maximum 20 different phases in one timing plan;

next_phase_id: the phase_id of the next phase. If it is NEMA phase plan, we will set the next_phase to "-1". And if it is not NEMA phase plan, the next phase is set to be a positive integer. In this example, we didn't use NEMA phase plan and there are only two phases (1 and 2) in the intersection;

green_duration: the green duration of this phase (does not consider the all red and yellow time);

movement_str: this is the most crucial element in this table. Here, the string format is like (**start_node_id**)_(**end_node_id**)_(**turn_direction**). **L** and **T** mean **left turn** and **through** respectively;

movement_dir_str: is the direction string of the movements, which we could get directly from input_movement.csv or from Synchro (will introduce in section 5). For example, WBL means west bound left turn.

Using the **input_timing.csv** file and **input_signal_timing_plan.csv** file, we could generate the **input_timing_status.csv** file with **timing_generator.exe** :

The **input_timing_status.csv** file looks like this:

Table 10. Sample data in file input_timing_status.csv

movement_str	start_time_in_sec	end_time_in_sec	signal_status	from_node_id	to_node_id	turn_type
1_3_T	1	120	1	1	5	T
1_2_L	1	120	1	1	5	L
3_1_T	1	120	1	3	5	T
3_4_L	1	120	1	3	5	L
2_4_T	121	190	1	2	5	T
2_3_L	121	190	1	2	5	L
4_2_T	121	190	1	4	5	T
4_1_L	121	190	1	4	5	L
…	…	…	…	…	…	…

start_time_in_sec: the start time of each small phase;

end_time_in_sec: the end time of each small phase;

signal_status: 1: green; 2: red;

from_node_id: from_node_id of this phase;

to_node_id: it represents the intersection id of this phase, (different from the movement_str in input_timing). We generate it in this way because, there are only real links between normal nodes and the intersection nodes. For example, when the movement string movement_str is 1_3_T, there is no direct link from node 1 to node 3, it is actually 1->5->3. Therefore, the to_node_id is 5 (the intersection id) and the turn type is T (Through).

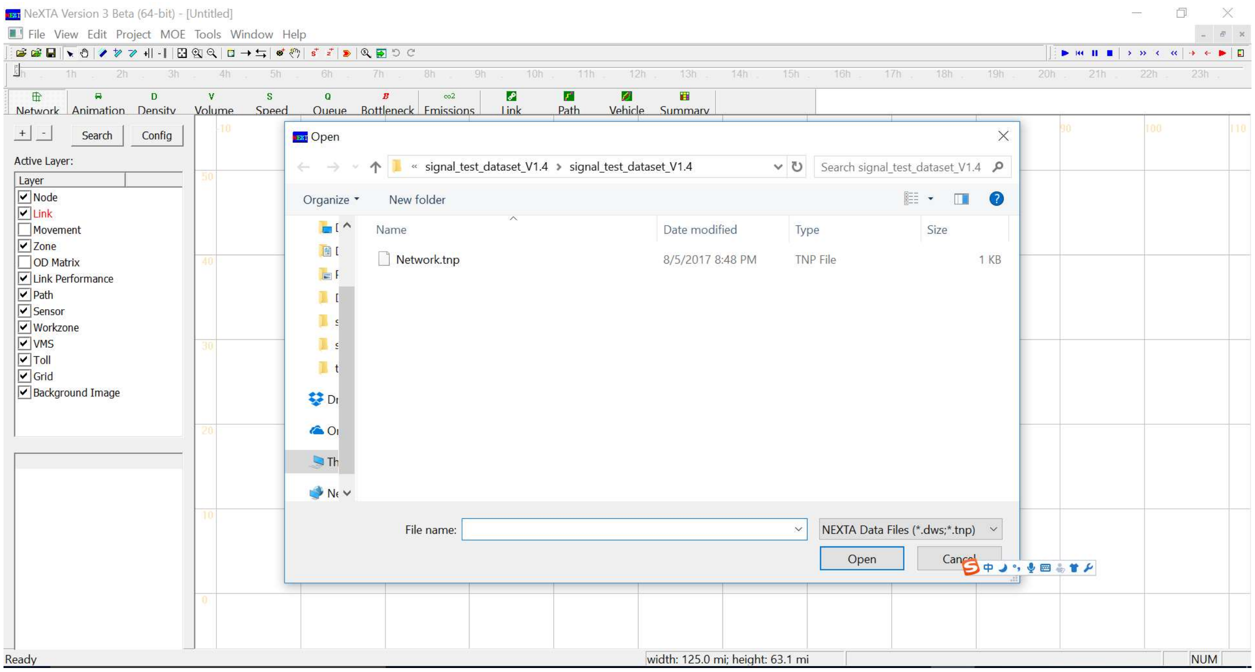
turn_type: the phase direction, **T** means through and **L** means left turn.

After preparing all these files, we copy them to the working folder with DTALite.exe and other input files.

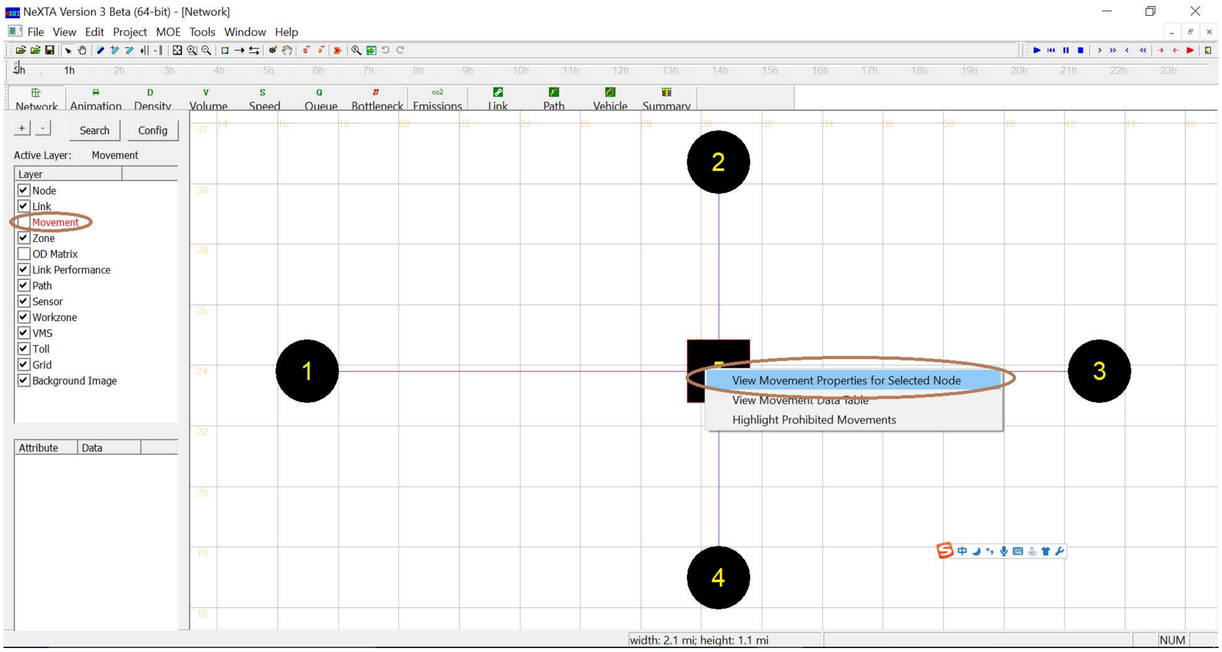
Section 4: NEXTA GUI instruction

The NEXTA GUI now could visualize the phase plans of each intersection.

Click NEXTA.exe and load the example project:

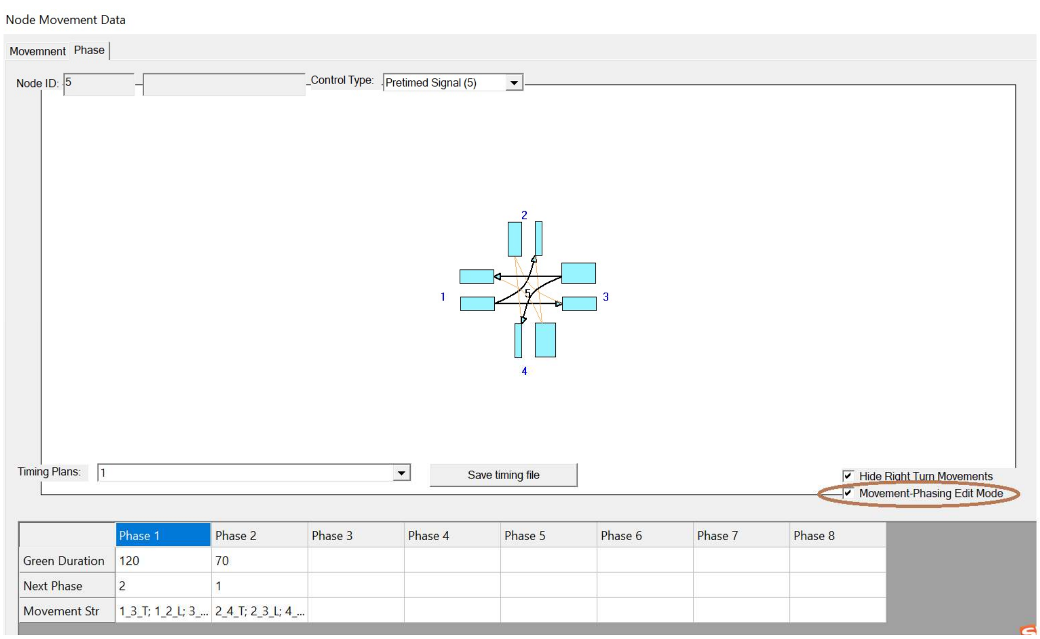


Activate the **Movement** layer -> right click the intersection node -> View Movement Properties for Selected Node.



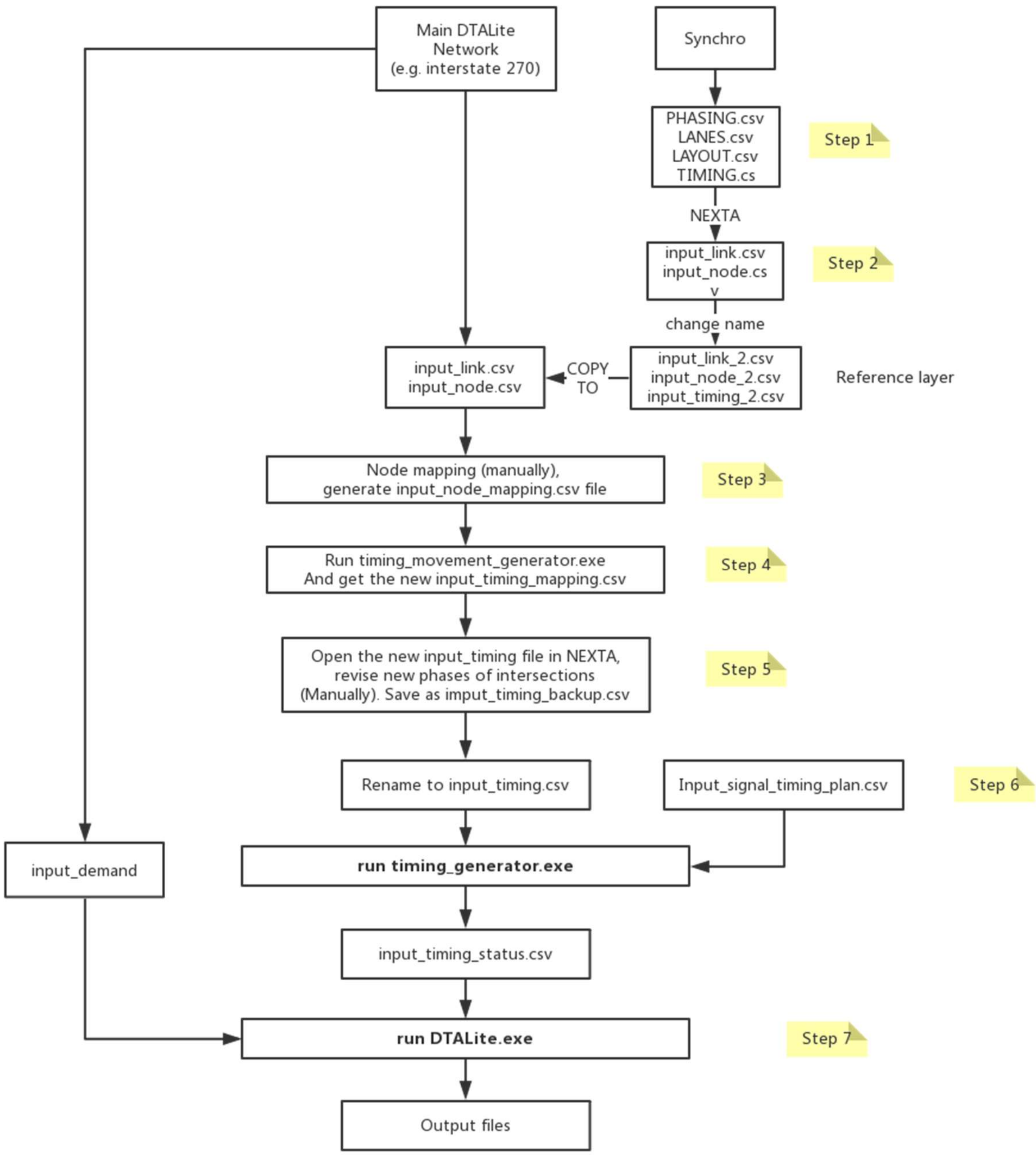
If this section already has the phase plans (the file input_timing.csv), we could display them in NEXTA.

To add a phase in current phase, we need to check the box next to **Movement-Phasing Edit Mode** on **"Phase"** tab in Node Movement Data. First, click the phase number in the table below that you wish to edit, then double click corresponding movement lines (orange) in the phase graph to include in the new phase. Then the movement selected will be added to the Movement_Str row in the table. You can remove movement segments by deleting text from the corresponding "Movement Str" cell. You can also change the "Green Duration" and/or "Next Phase" values in the table (for NEMA phase format, it is all -1 and do not need to change them). By clicking the "**Save Timing File**" and **"OK"** button, the **input_timing.csv** file will automatically update in the same folder.



Section 5: Step by step instruction of how to import data from Synchro to DTALite

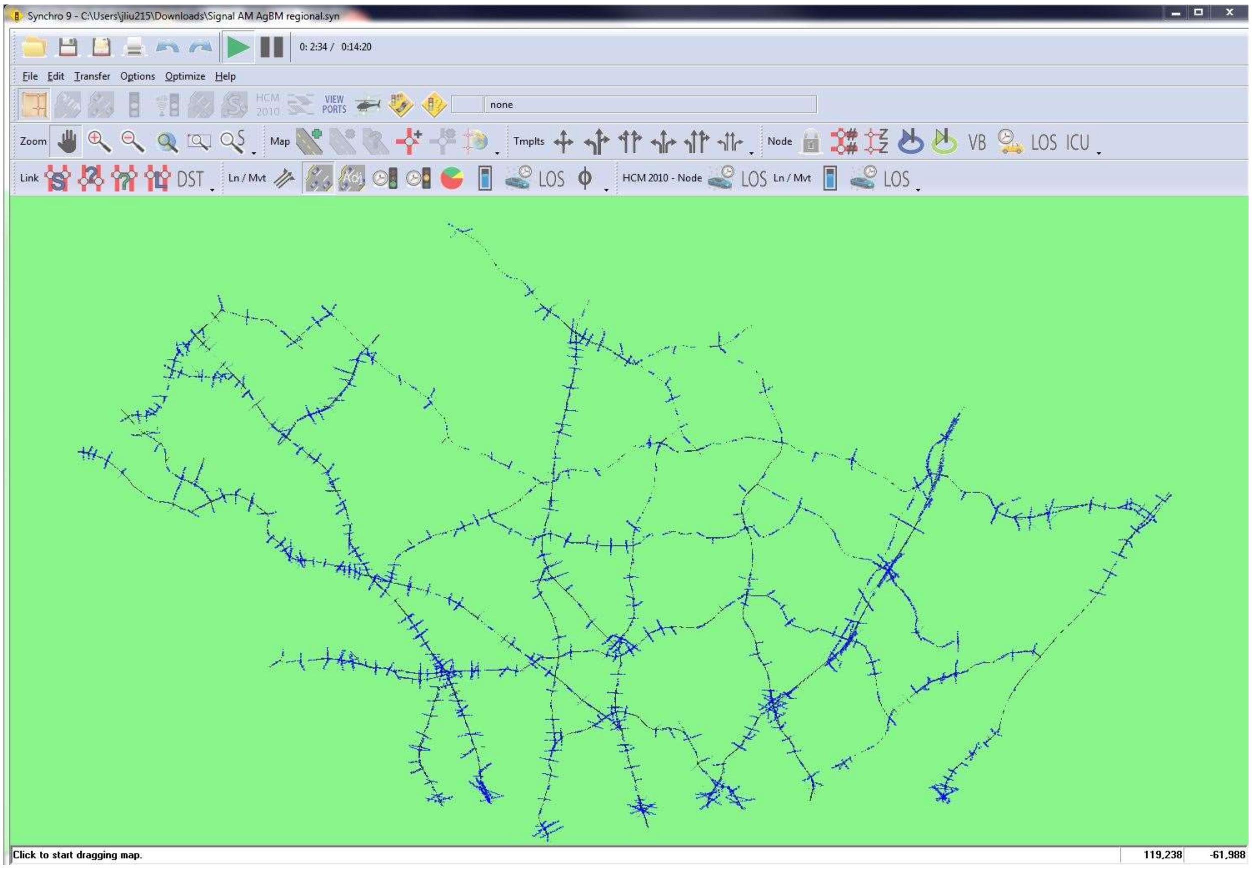
The flow chart of a seven-step procedure of synchro data importing function:



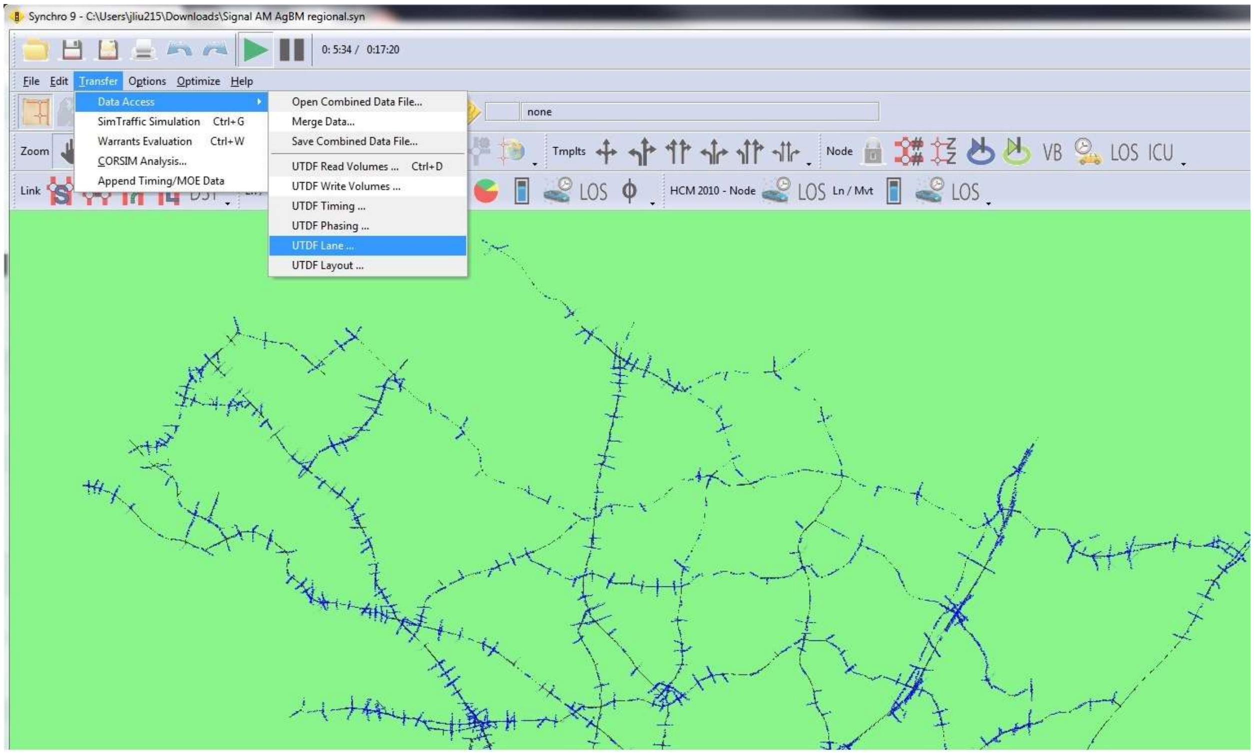
Step 1: Export csv files from Synchro

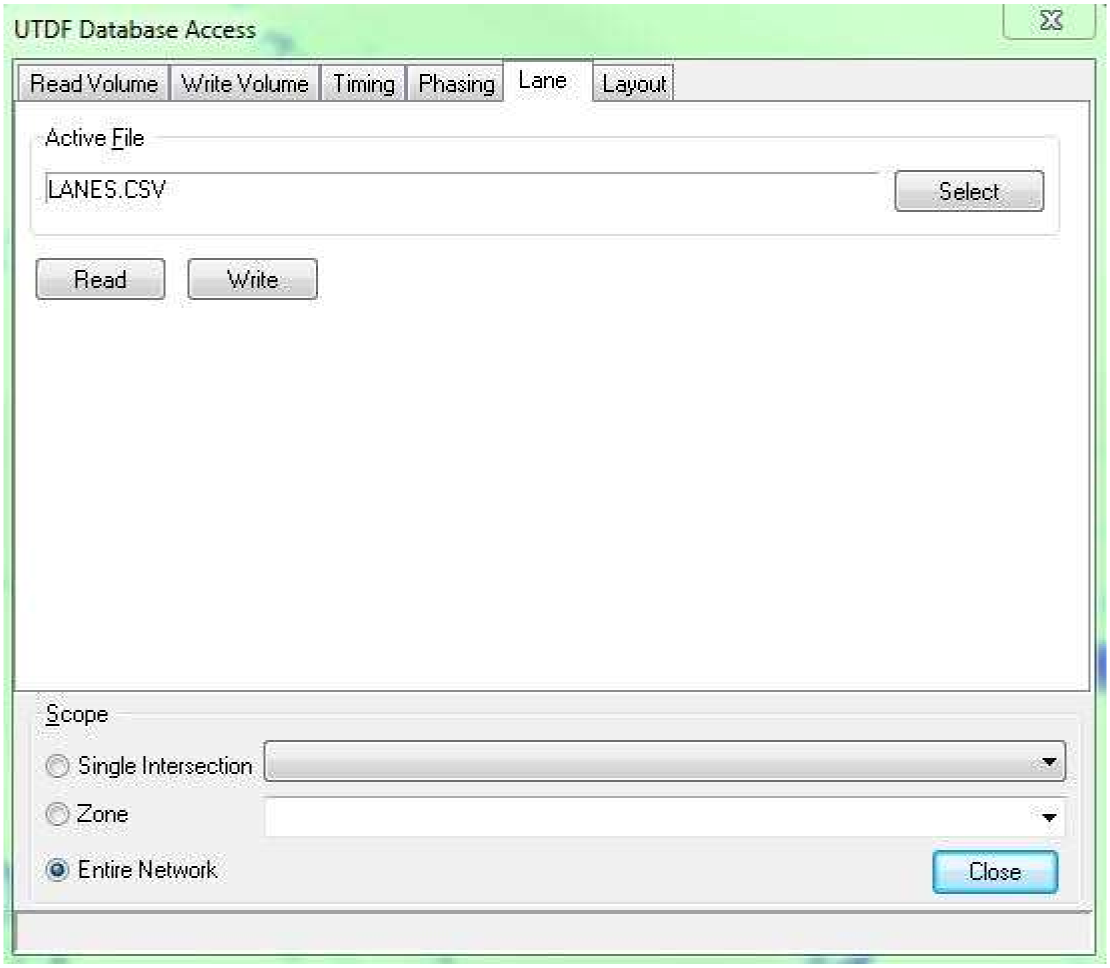
Prepare your Synchro dataset. Currently, *only Synchro data can be analyzed with DTALite.*

Note: Sample Synchro data has been provided and stored in the **1-Signal Timing Data_from Synchro** folder. Sample Synchro data has already been exported and saved in **2-Interstate_270_synchro_export** folder. **You may skip ahead to step 2 if using the provided sample data**.

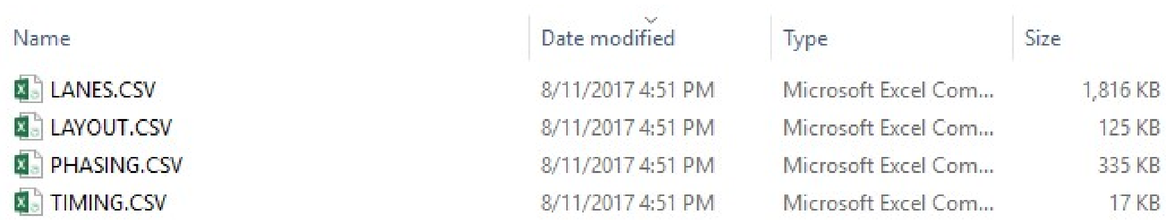


Next, export the data to LANES.csv, LAYOUT.csv, PHASING.csv and TIMING.csv (UTDF format) as shown below.


Synchro 9 - C:\Users\jliu215\Downloads\Signal AM AgBM regional.syn
0: 5:34 / 0:17:20
File Edit Transfer Options Optimize Help
Data Access
SimTraffic Simulation Ctrl+G
Warrants Evaluation Ctrl+W
CORSIM Analysis...
Append Timing/MOE Data
Open Combined Data File...
Merge Data...
Save Combined Data File...
UTDF Read Volumes ... Ctrl+D
UTDF Write Volumes ...
UTDF Timing ...
UTDF Phasing ...
UTDF Lane ...
UTDF Layout ...
none
Zoom
Link
Tmplts
Node
VB
LOS
ICU
HCM 2010 - Node
Ln / Mvt



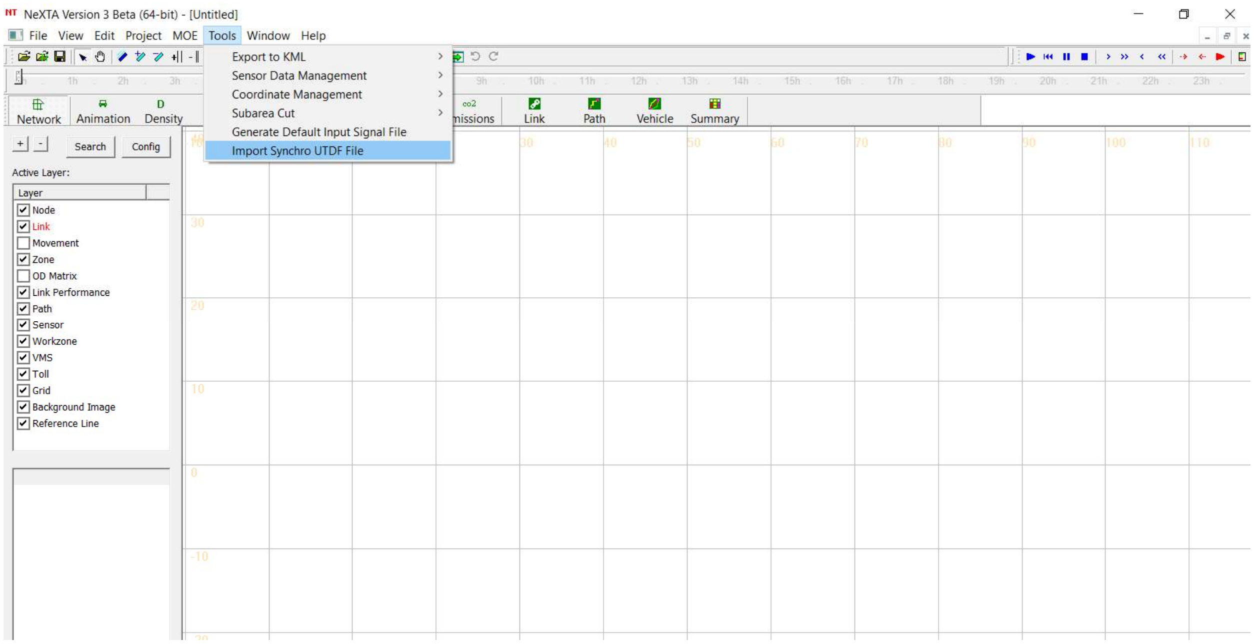
Once exported, all four Synchro file exports should appear in **2-Interstate_270_synchro_export** folder (shown below).



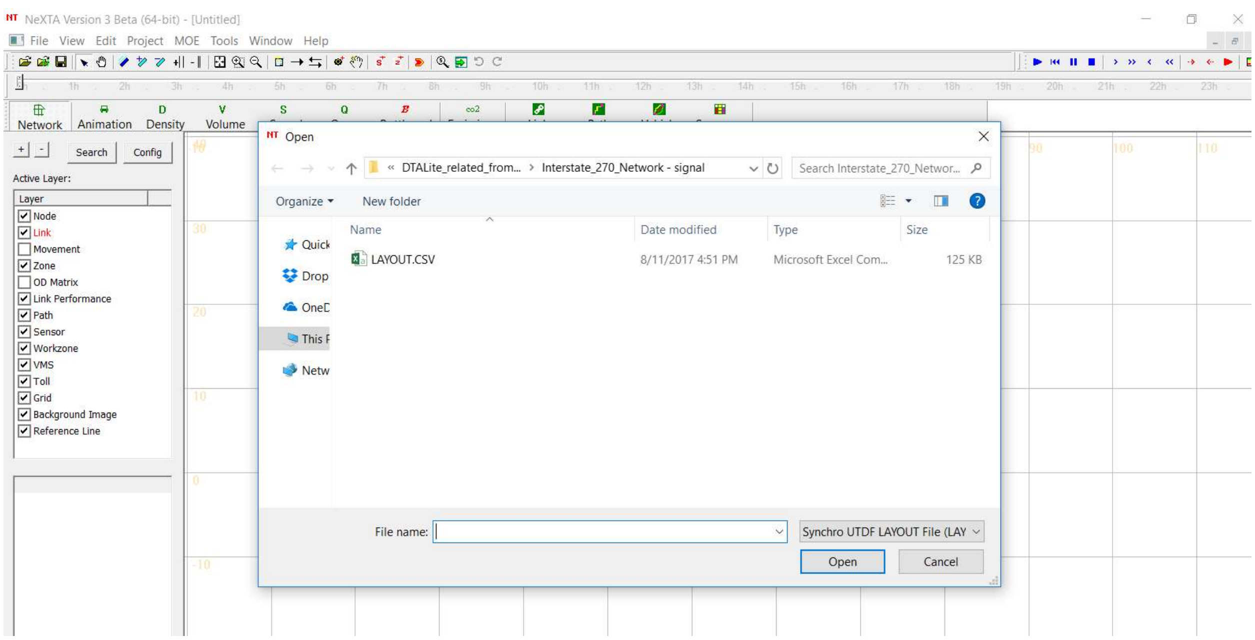
Name	Date modified	Type	Size
LANES.CSV	8/11/2017 4:51 PM	Microsoft Excel Com...	1,816 KB
LAYOUT.CSV	8/11/2017 4:51 PM	Microsoft Excel Com...	125 KB
PHASING.CSV	8/11/2017 4:51 PM	Microsoft Excel Com...	335 KB
TIMING.CSV	8/11/2017 4:51 PM	Microsoft Excel Com...	17 KB

Step 2: Use NEXTA to import the Synchro UTDF format

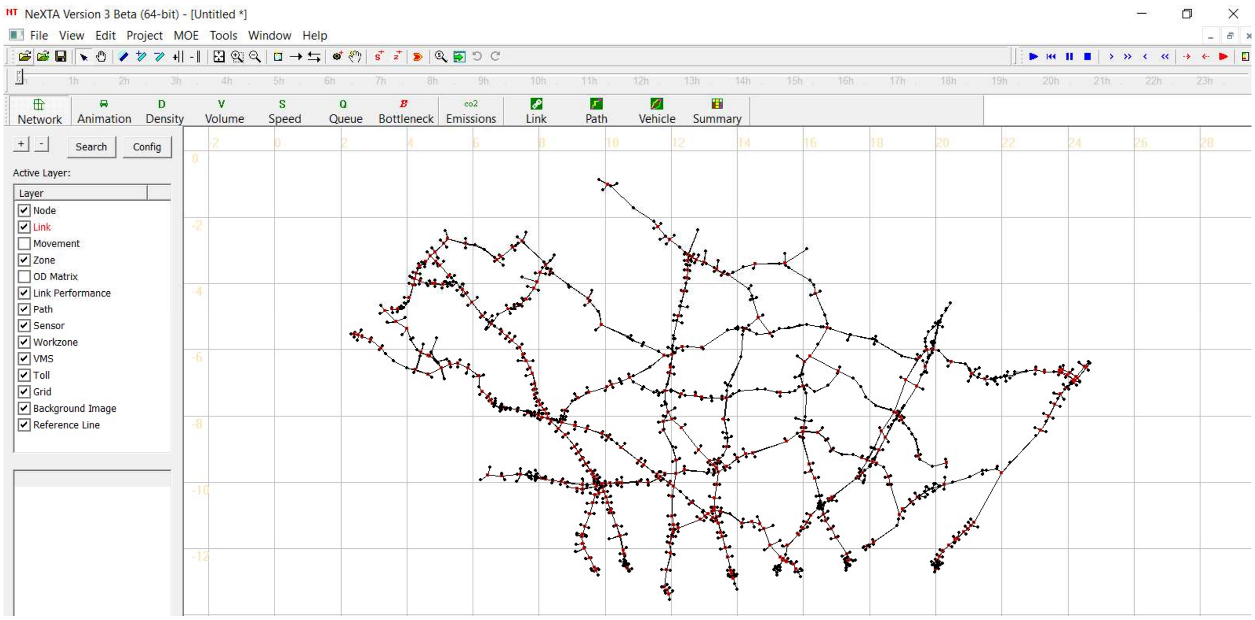
Open NEXTA in **2-Interstate_270_synchro_export**, click tools -> Import Synchro UTDF File:



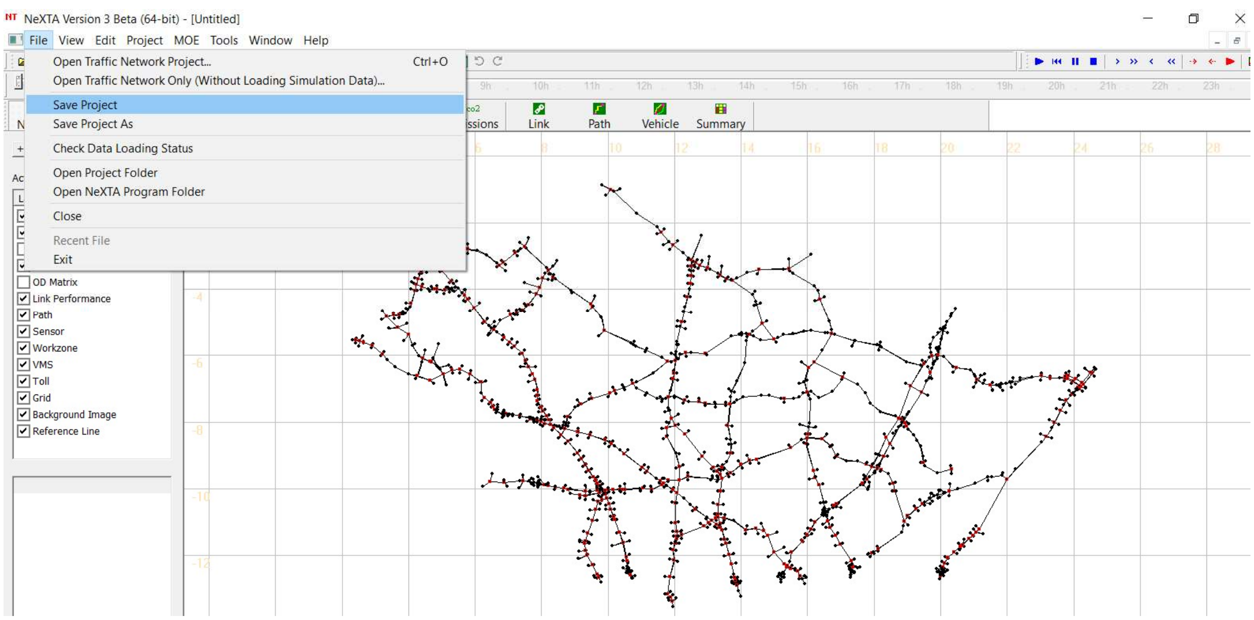
Open LAYOUT.csv file that saved above:

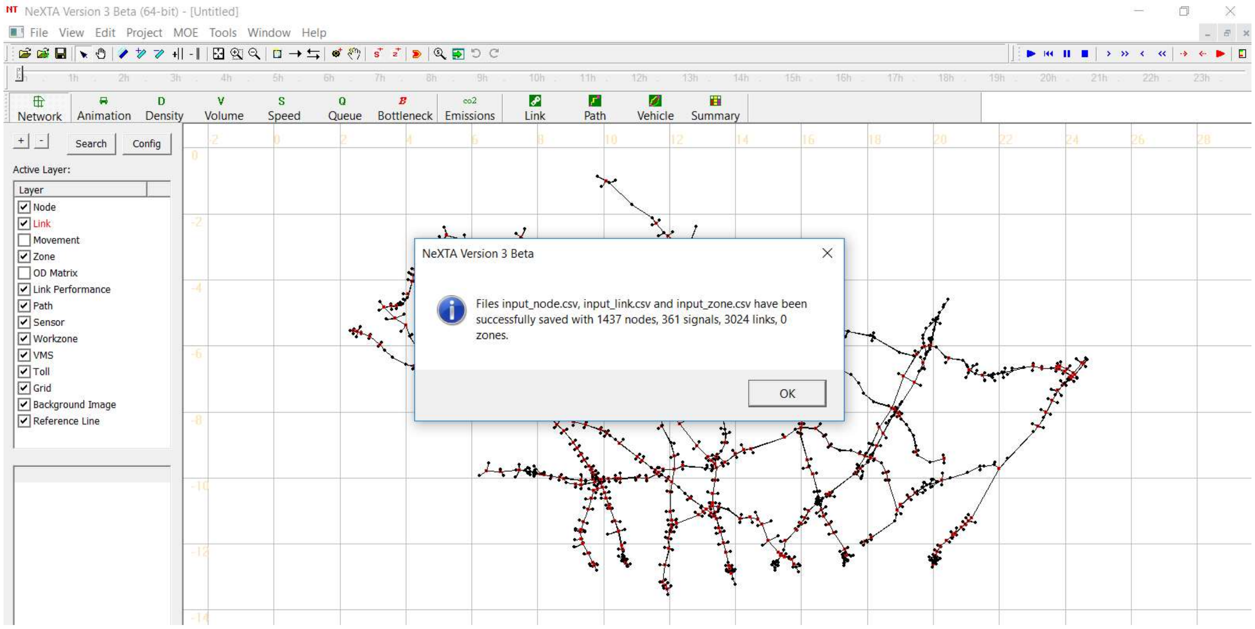


The synchro data will be imported to NEXTA.



Save the project in the same **2-Interstate_270_synchro_export** folder.





The network and the signal data will be automatically populated in the same folder.

Name	Date modified	Type	Size
input_activity_location.csv	8/13/2017 9:30 PM	Microsoft Excel Com...	1 KB
input_demand_type.csv	8/13/2017 9:30 PM	Microsoft Excel Com...	1 KB
input_link_type.csv	8/13/2017 9:30 PM	Microsoft Excel Com...	1 KB
input_timing_backup.csv	8/13/2017 9:30 PM	Microsoft Excel Com...	76 KB
input_zone.csv	8/13/2017 9:30 PM	Microsoft Excel Com...	1 KB
output_zone.csv	8/13/2017 9:30 PM	Microsoft Excel Com...	1 KB
Scenario_Dynamic_Message_Sign.csv	8/13/2017 9:30 PM	Microsoft Excel Com...	1 KB
Scenario_Link_Based_Toll.csv	8/13/2017 9:30 PM	Microsoft Excel Com...	1 KB
Scenario_Work_Zone.csv	8/13/2017 9:30 PM	Microsoft Excel Com...	1 KB
synchro_270.tnp	8/13/2017 9:30 PM	TNP File	1 KB
input_link.csv	8/13/2017 9:30 PM	Microsoft Excel Com...	827 KB
input_movement.csv	8/13/2017 9:30 PM	Microsoft Excel Com...	306 KB
input_node.csv	8/13/2017 9:30 PM	Microsoft Excel Com...	167 KB
input_node_control_type.csv	8/13/2017 9:30 PM	Microsoft Excel Com...	1 KB
LANES.CSV	8/11/2017 4:51 PM	Microsoft Excel Com...	1,816 KB
LAYOUT.CSV	8/11/2017 4:51 PM	Microsoft Excel Com...	125 KB
PHASING.CSV	8/11/2017 4:51 PM	Microsoft Excel Com...	335 KB
TIMING.CSV	8/11/2017 4:51 PM	Microsoft Excel Com...	17 KB
output_LinkMOE.csv	7/10/2017 2:34 PM	Microsoft Excel Com...	538 KB
output_LinkTDMOE.bin	7/10/2017 2:34 PM	BIN File	44,958 KB
agent.bin	7/10/2017 2:34 PM	BIN File	75,380 KB
input_scenario_settings.csv	6/28/2017 10:33 AM	Microsoft Excel Com...	1 KB
demand.dat	10/13/2014 2:59 AM	DAT File	189 KB
demand_HOV.dat	10/13/2014 2:59 AM	DAT File	189 KB
demand_truck.dat	10/13/2014 2:59 AM	DAT File	189 KB

Step 3: Mapping two networks

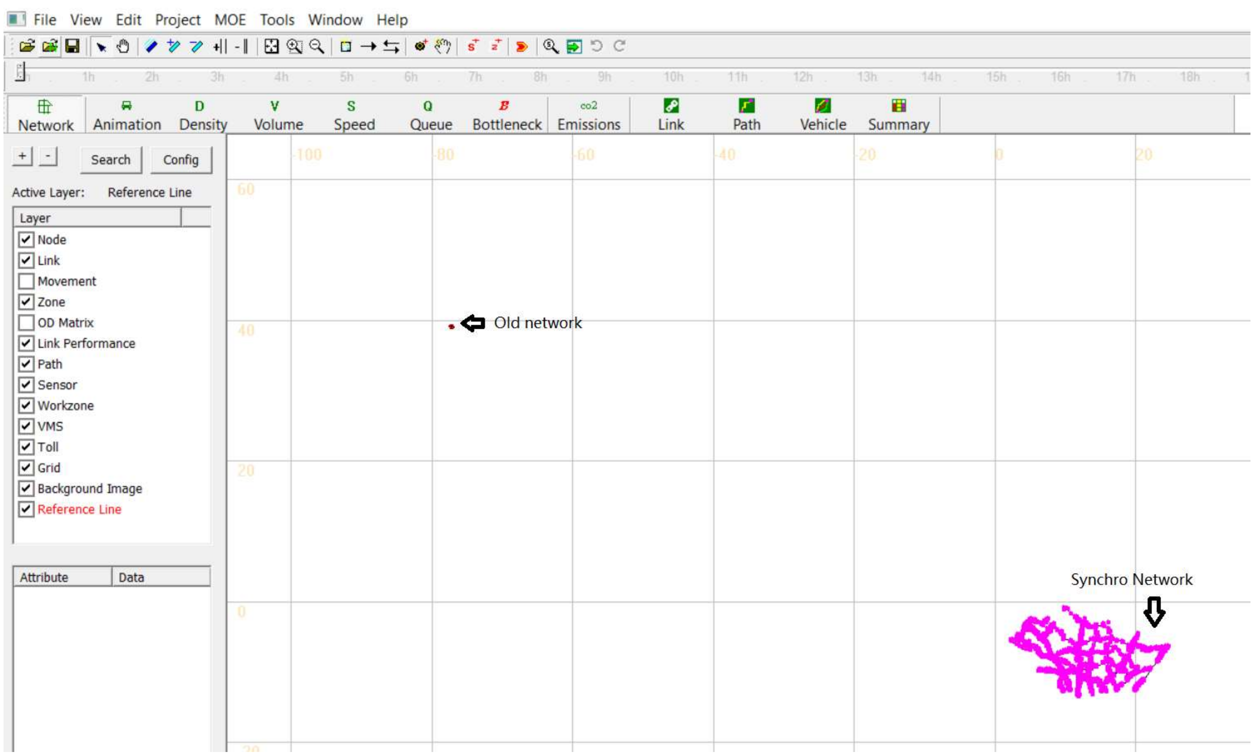
In this step, we need two networks:

(a) **Network 1:** DTALite main network
(in **3-Interstate_270_Network - mapping\Main_DTALite_Network**)

(b) **Network 2**: the exported synchro network in DTALite format
(in **2-Interstate_270_synchro_export**)

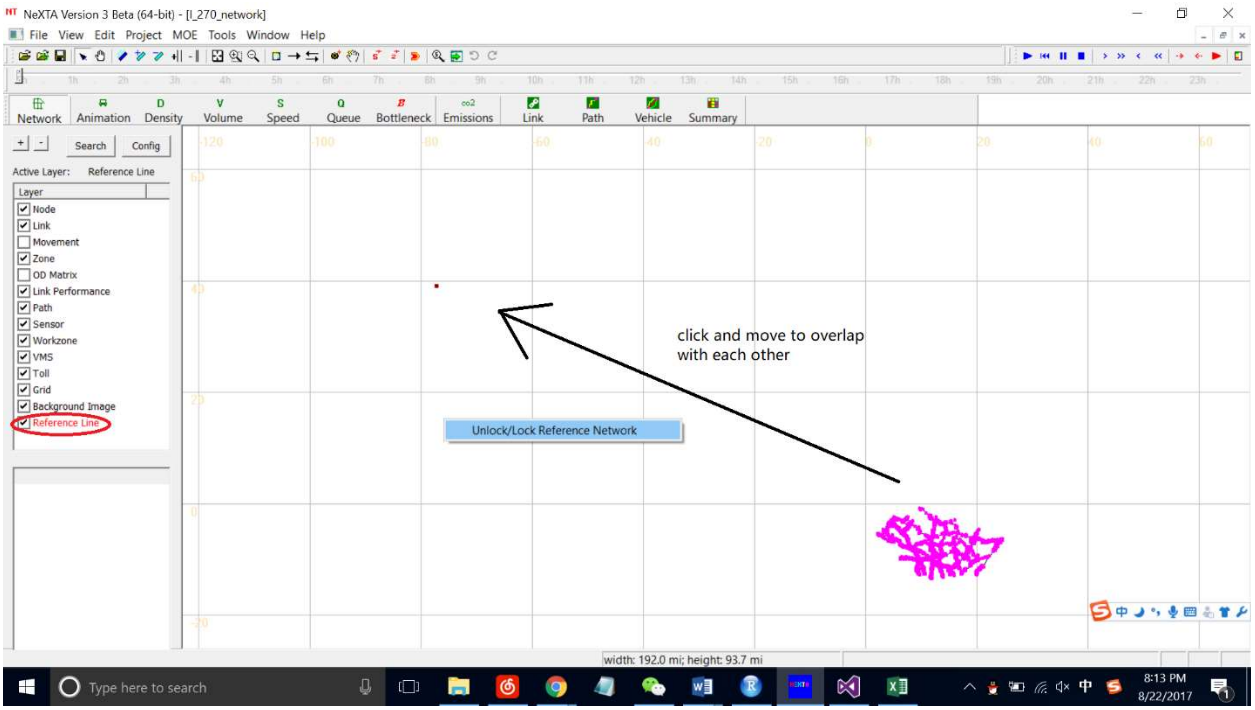
Here, the goal is to overlay both networks to identify and match the node IDs for all intersections to be analyzed. Use NEXTA to spatially reference (resize and shift) **Network 2** with **Network 1** in order to later join both networks' signal data (originally in different coordinate system). NEXTA has a tool to display all joined intersections to verify the results. Note: you don't need to join all the intersection nodes, just the intersections that you wish to analyze using the main DTALite network.

Rename the **input_node.csv** and **input_link.csv** in **Network 2** to **input_node_2.csv** and **input_link_2.csv.** Copy both to the main DTALite **Network 1** (in **3-Interstate_270_Network – mapping \Main_DTALite_Network**). Then, open the new project in NEXTA.

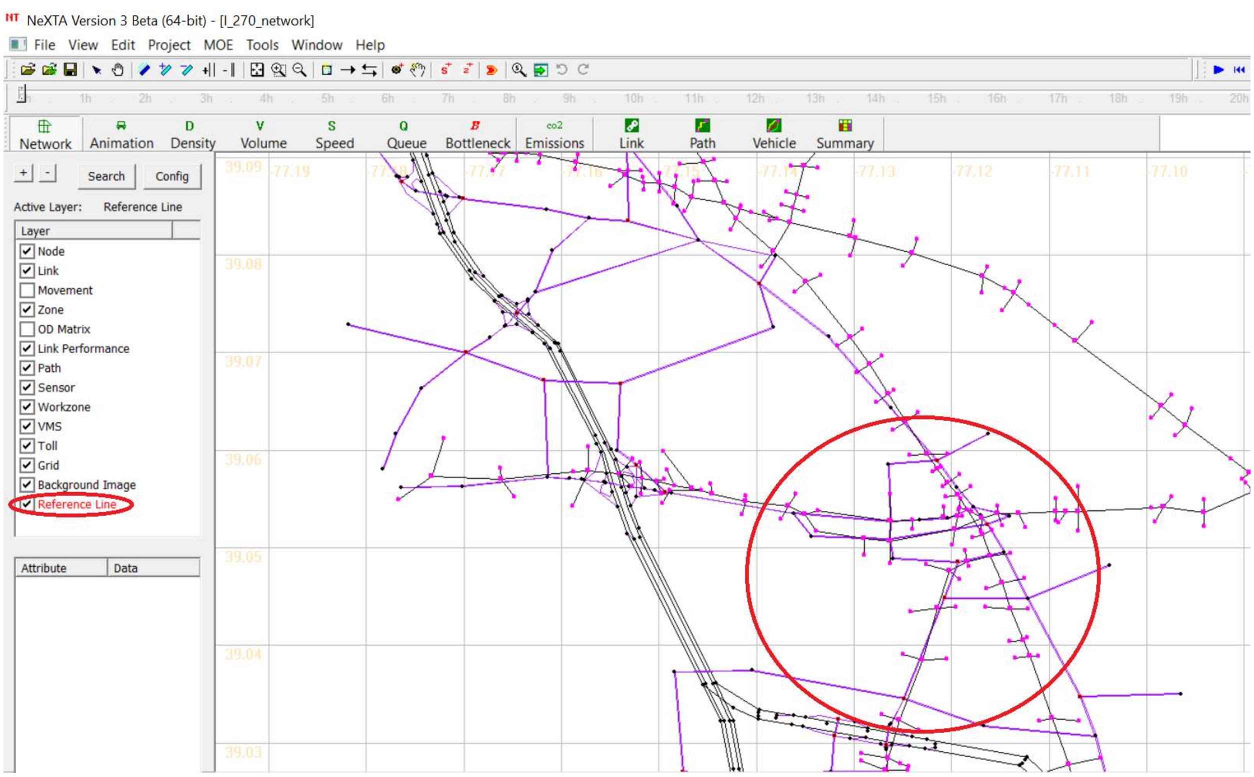


Note: Both networks won't initially display together on screen at first. User will have to zoom in and out to search for both networks.

Activate **reference line** "layer", right click inside the grid to unlock the reference network (**Network 2**).

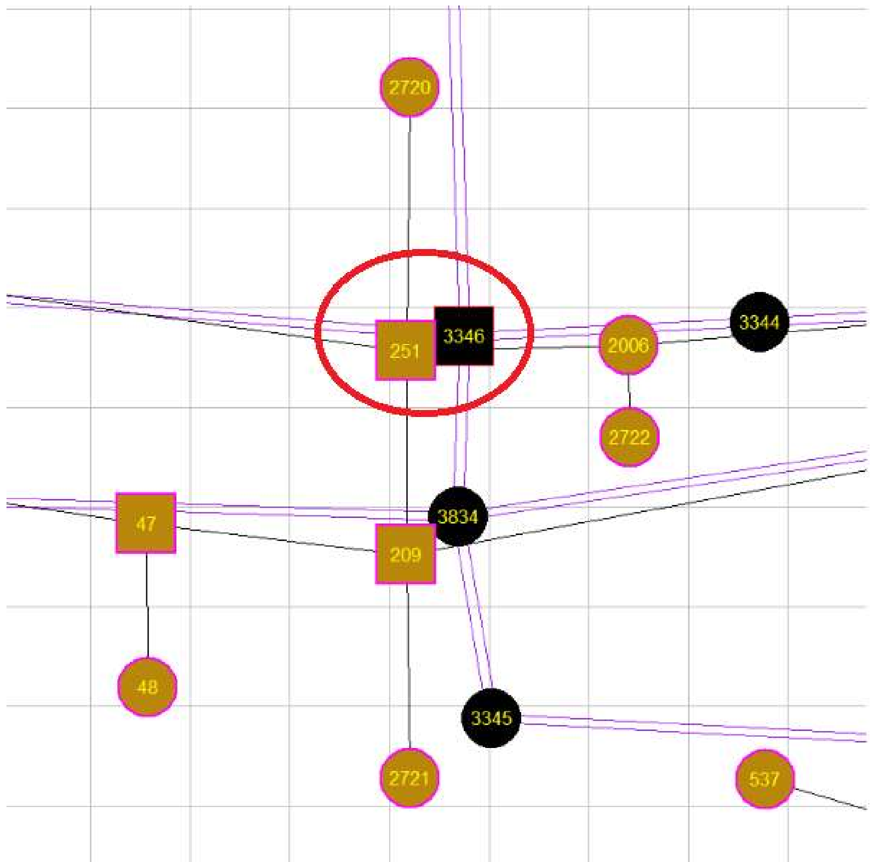


Move the two networks to match with each other. Click and drag mouse to move; scroll in/out to zoom in/out. (**Be careful**: don't close the NEXTA GUI when you are matching these two networks, otherwise you need to do it again)



It could be difficult to accurately overlay the two networks together because the Synchro network was mapped in more detail and includes more intersections. Therefore, you may have to pick an intersection id from Network 2 and identify its street name from the Synchro **LAYOUT.csv** file and use satellite imagery (Google Maps) to identify the relative location of these intersections.

Once able to spatially overlay a pair of intersections, the **node and Link ID** of both networks will need to be matched.



To match intersection IDs, create a **.csv** file named “**input_node_mapping”** in the **3-Interstate_270_Network** folder. Within the .csv file create the fields “**node_id_1**” and “**node_id_2**” and manually code the respective intersection nodes for each field. **Node_id_1** should contain the node IDs from the main DTALite network (black); **node_id_2** should contain the matching nodes from the Synchro exported network (orange). Only code the intersections you want DTALite to analyze.

Input_node_mapping.csv

node_id_1	**node_id_2**
3346	251
3335	1208
…	…

Step 4: Generate the new input_timing_mapping.csv

Copy the **input_node_mapping**.**csv** file created in step 3 to the **4-input_timing_movement_generator folder**.

Now copy the **input_timing.csv** from the main DTALite network (3-Interstate_270_Network - mapping\Main_DTALite_Network) to the **4-input_timing_movement_generator** folder. Copy the **input_timing.csv** (2-Interstate_270_synchro_export) to the **4-input_timing_movement_generator** folder and rename it as **input_timing_2.csv**. Open both **input_timing.csv** files once copied to the **4-input_timing_movement_generator** folder to ensure they look display as follows:

input_timing.csv

int_id	timing_plan_no	phase_id	next_phase_id	green_duration	movement_str	movement_dir_Str
3724	1	1	2	60	;73_80_T;76_80_L	
3724	1	2	1	60	;73_75_L;74_75_T	
3724	2	1	2	69	;73_80_T;73_75_L;76_80_L	
3724	2	2	1	69	;74_75_T	

Part of **input_timing_2.csv**

int_id	timing_plan_no	phase_id	next_phase_id	green_duration	movement_str	movement_dir_str
251	1	1		33	;2503_2720_L	EBL;
251	1	2		50	;2006_2503_T	WBT;
251	1	4		18	;209_2720_T;209_2503_L	NBT;NBL;
251	1	5		59	;2006_209_L	WBL;
251	1	6		24	;2503_2006_T	EBT;
251	1	8		23	;2720_209_T;2720_2006_L;2720_2503_R	SBT;SBL;SBR;
1208	1	1		7	;1209_531_L	SBL;
1208	1	2		72	;1207_1209_T;1207_532_L;1207_531_R	NBT;NBL;NBR;
1208	1	4		30	;532_531_T;532_1209_L	EBT;EBL;
1208	1	5		16	;1207_532_L	NBL;
1208	1	6		62	;1209_1207_T;1209_531_L	SBT;SBL;
1208	1	8		18	;531_532_T;531_1207_L;531_1209_R	WBT;WBL;WBR;

Next, run the **timing_movement_generator.exe** application to generate the new **input_timing_mapping.csv** that contains the combined phasing plan for both networks. Open the **input_timing_mapping.csv** file.

input_timing_mapping.csv

int_id	timing_plan_no	phase_id	next_phase_id	green_duration	movement_str	movement_dir_str
3724	1	1	2	60	;73_80_T;76_80_L	
3724	1	2	1	60	;73_75_L;74_75_T	
3724	2	1	2	69	;73_80_T;73_75_L;76_80_L	
3724	2	2	1	69	;74_75_T	
3346	1	1	-1	33		EBL;
3346	1	2	-1	50		WBT;
3346	1	4	-1	18		NBT;NBL;
3346	1	5	-1	59		WBL;
3346	1	6	-1	24		EBT;
3346	1	8	-1	23		SBT;SBL;SBR;
3335	1	1	-1	7		SBL;
3335	1	2	-1	72		NBT;NBL;NBR;
3335	1	4	-1	30		EBT;EBL;
3335	1	5	-1	16		NBL;
3335	1	6	-1	62		SBT;SBL;
3335	1	8	-1	18		WBT;WBL;WBR;

If the "next_phase_id" field within the **input_timing_mapping.csv** file displays any empty cells, input"-1" as the next_phase_id. In the next step to generate **input_timing_status.csv**, we will use the next_phase_id values to distinguish the different **NEMA** signal phase plans .

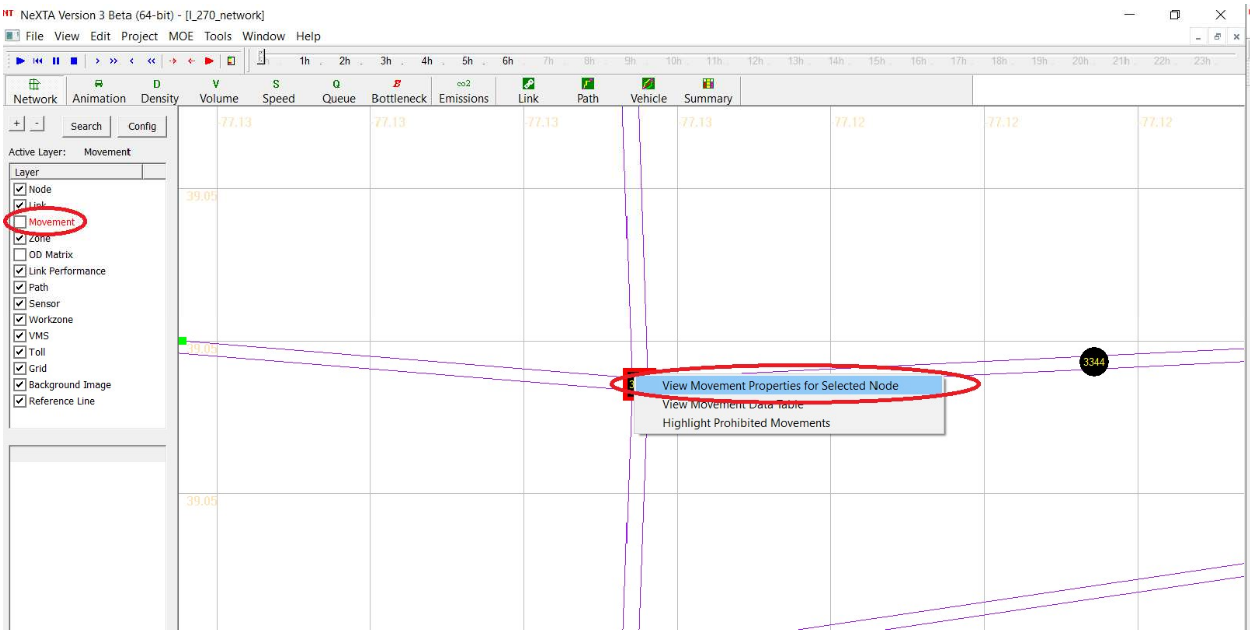
input_node_mapping.csv	8/17/2017 6:12 PM	Microsoft Excel Co...	1 KB
input_timing.csv	8/19/2017 3:55 PM	Microsoft Excel Co...	1 KB
input_timing_2.csv	8/13/2017 9:30 PM	Microsoft Excel Co...	76 KB
input_timing_mapping.csv	8/19/2017 4:04 PM	Microsoft Excel Co...	1 KB

Your **4-input_timing_movement_generator** folder should now include the above files.

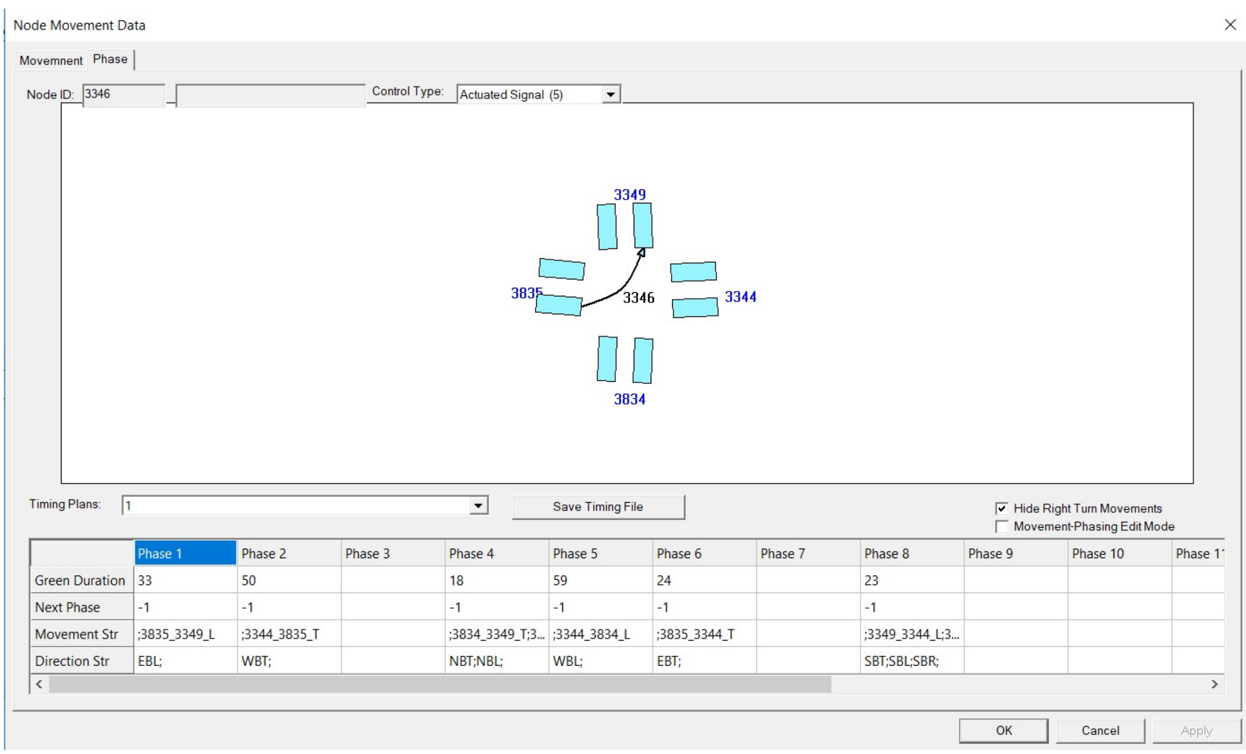
Step 5: Verify the input phasing results

Open **5-Interstate_270_Network - Phase Based Signal Representation Model** (this is a copy of the main DTALite Network in step 3) and copy the **input_timing_mapping.csv** generated in step 4 to this folder, rename as **input_timing.csv.** If you open this file, you should notice that the movement string (**movement_str**) values are not populated for all the intersection nodes sourced from Synchro. However, NEXTA will generate the movement strings (**movement_str**) automatically.

First, open the **I_270_network.tnp** file (**5-Interstate_270_Network - Phase Based Signal Representation Model)** in NEXTA. Turn on "Movement" layer. Zoom to one of the signalized intersections (for example intersection 3335) that was previously matched. Right click on the intersection node and click "View Movement Properties for Selected Node".



Click the “Phase” tab and validate the accuracy of the phase data. Make phase adjustments if needed (see section 4 for more details on how to edit phase data). Once the data is validated, click” **Save Timing File**” and “**OK”.**



Now the **input_timing.csv** should display the updated movement strings (see table below).

int_id	timing_plan_no	phase_id	next_phase_id	green_duration	movement_str	movement_dir_str
3335	1	1	-1	7	;3337_3359_L	SBL;
3335	1	2	-1	72	;156_155_R;156_3336_L;156_3337_T;156_3359_R	NBT;NBL;NBR;
3335	1	4	-1	30	;3336_3337_L;3336_3359_T	EBT;EBL;
3335	1	5	-1	16	;156_3336_L	NBL;
3335	1	6	-1	62	;3337_155_T;3337_3359_L	SBT;SBL;
3335	1	8	-1	18	;3359_155_L;3359_3336_T;3359_3337_R	WBT;WBL;WBR;
3346	1	1	-1	33	;3835_3349_L	EBL;
3346	1	2	-1	50	;3344_3835_T	WBT;
3346	1	4	-1	18	;3834_3349_T;3834_3835_L	NBT;NBL;
3346	1	5	-1	59	;3344_3834_L	WBL;
3346	1	6	-1	24	;3835_3344_T	EBT;
3346	1	8	-1	23	;3349_3344_L;3349_3834_T;3349_3835_R	SBT;SBL;SBR;
3724	1	1	2	60	;73_80_T;76_80_L	
3724	1	2	1	60	;73_75_L;74_75_T	
3724	2	1	2	69	;73_80_T;73_75_L;76_80_L	
3724	2	2	1	69	;74_75_T	

Furthermore, open the **input_signal_timing_plan.csv** file and manually code the timing plans for each intersection listed in the **input_timing.csv** file. Please find table 8 in section 3 for more detailed information. The "**int_id**" and "**timing_plan_no**" should be matched with these in **input_timing.csv** table. And other attributes should manually code based on experience or other sourced data.

input_signal_timing_plan.csv

[TOD]	int_id	timing_plan_no	start_time_in_sec	end_time_in_sec	starting_phase	off_set
	3346	1	0	86400	1	11
	3335	1	0	86400	1	11
	3724	1	0	40000	1	11
	3724	2	40000	86400	1	11

At last, please make sure that all the intersection nodes are set to signalized intersection (control_type = 5) in input_node.csv:

input_node.csv

name	node_id	control_type	control_type_name	cycle_leng
	3335	5	actuated_signal	0
	3346	5	actuated_signal	0
	3724	5	actuated_signal	0

Step 6: Generate input_timing_status.csv file

Copy the **timing_generator.exe** to **5-Interstate_270_Network - Phase Based Signal Representation Model** folder. The **timing_generator.exe** could generate the **input_timing_status.csv** for both NEMA phase data and other phase data. However, make sure all the "**next_phase_id**" values in the **input_timing.csv** are present. For the NEMA phase plans, all the **next_phase_id** should be set to "-1".

Furthermore, make sure that the **input_timing.csv** and **input_signal_timing_plan.csv** have the same number of signal intersections and same timing plans for each intersection (for example, they all have three intersections with id: 3335, 3346 and 3724; and each intersection has the same number of timing plans).

Then, run the **timing_generator.exe** to generate the **input_timing_status.csv** file which will be used in **DTALite.exe** for simulation.

Step 7: Run DTALite.exe

Make sure all values under the "**signal_representation_model"** field in the **input_scenario_settings.csv** file is set to the right number, "2" for **Phase Based Signal Representation Model** which we mainly discussed in the documentation.

As an example, we set a simple demand as:

input_demand.csv

from_zone_id	to_zone_id	number_of_trips_demand_type1
5	6	100

Make sure the input_demand_file_list.csv file is set to be the right demand file as **input_demand.csv**:

scenario_no	file_sequence_no	file_name	format_type
0	1	input_demand.csv	column

Then, run DTALite.exe to get the output results.

Step 8: Check the output results

(It is an optimal step to understand the outputs)

Open **output_agent.csv** and go to **path_node_sequence** and **path_time_sequence** column inside **5-Interstate_270_Network - Phase Based Signal Representation Model**.

Find one path which include the signalized intersections. Filter one of the same path out and find the intersection and upstream node of this intersection, here we use 3335 (intersection) and 3336 (upstream node). The **path_time_sequence** (unit in minute) is also showed in the table. Also, the free flow travel time could be got from **output_linkMOE.csv** from node 3336 to node 3335, which is 1.71 minutes.

output_agent.csv

path_node_sequence	path_time_sequence
3336;3335;3337;	62.90;64.90;66.20;
3336;3335;3337;	65.90;68.50;69.80;
3336;3335;3337;	68.90;70.62;71.90;

output_linkMOE.csv

from_node_id	to_node_id	free-flow_travel_time
3336	3332	0.8542
3336	3335	1.7121

- time leaving node 3336: 62.9*60 = 3774 sec
- time arriving at node 3335 under free-flow travel time: (62.9+1.71) *60 = 3876.6 sec
- time leaving node 3335: 64.9*60 = 3894 sec

And from the input_timing_status.csv file, the start green time (start_time_in_sec) of movement 3336_3337_L is 3895 second, which is exactly the time when this agent is going to leave intersection 3335.

input_timing_status.csv

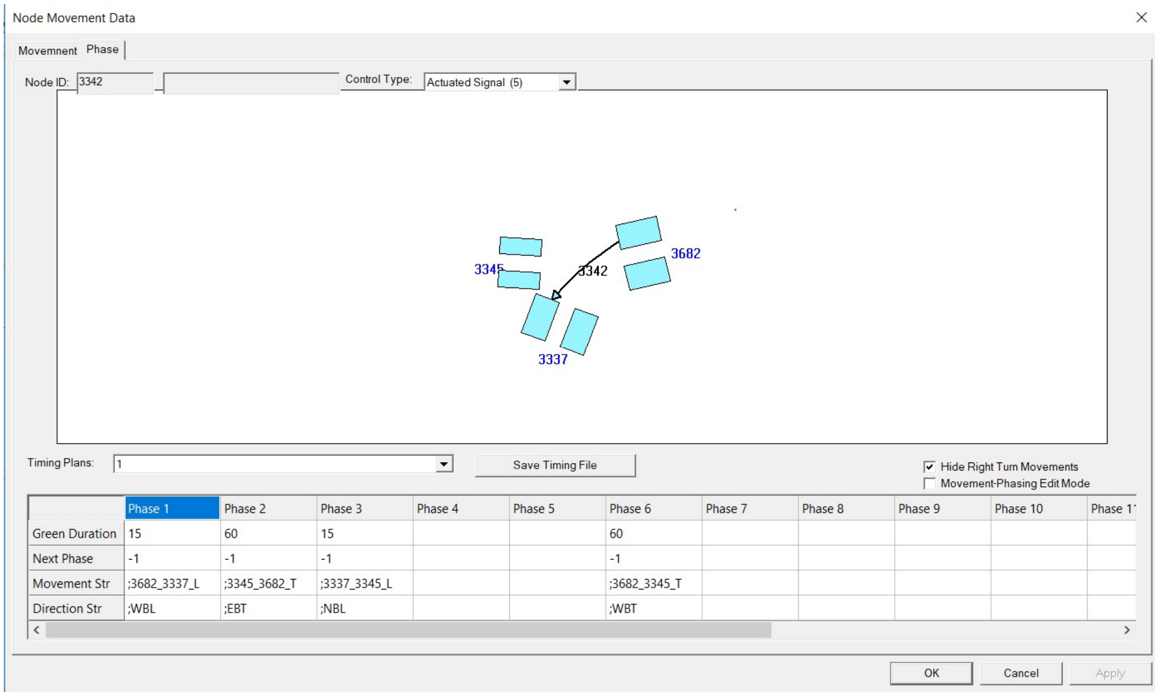
movement_str	start_time_in_s	end_time_in_sec	signal_status	from_node_id	to_node_id	turn_type
3336_3337_L	3786	3815	1	3336	3335	L
3336_3337_L	3895	3924	1	3336	3335	L

Section 6: Save default signal plan

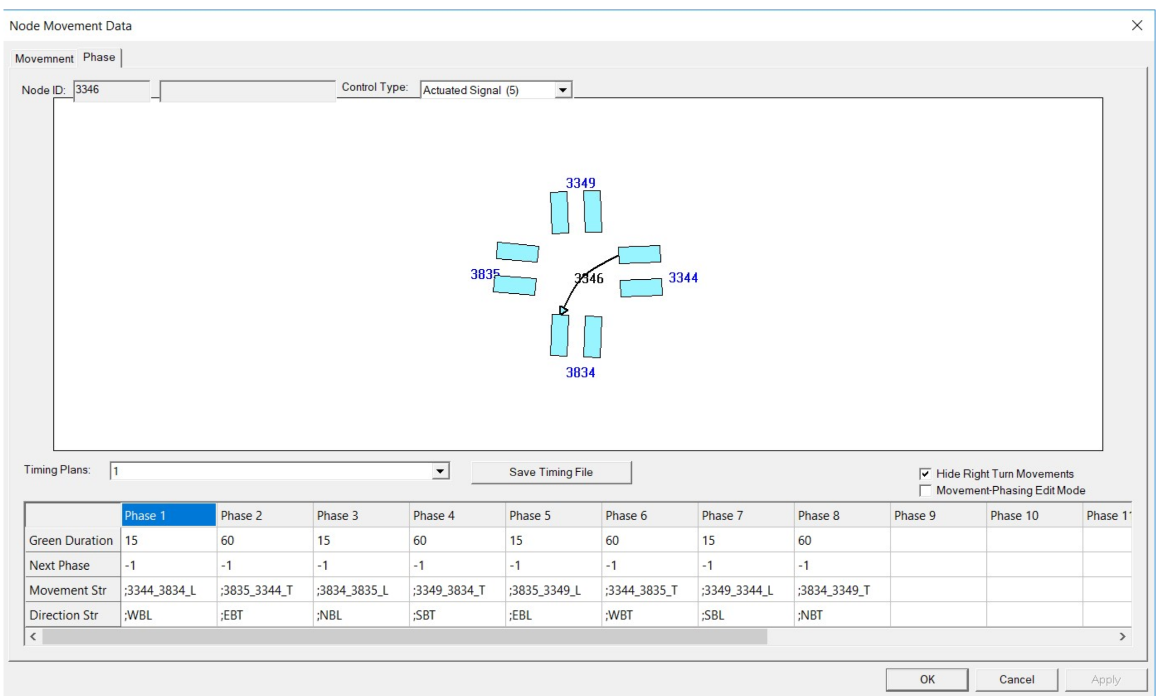
In the previous sections, all the intersection signal timing data is either manually added with signal plans or imported from Synchro data. In the new NEXTA.exe, now we can generate the default phase plans.

For example, open **Section 6 - Main_DTALite_Network - Default Signal Plan**. This file was built by copying main DTALite network in the **3-Interstate_270_Network - mapping\Main_DTALite_Network**, and save as **Section 6 - Main_DTALite_Network - Default Signal Plan** and delete the **input_timing.csv, input_node_2 and input_link_2** file inside it. When we reopen the network in NEXTA, the signal plans of these intersections will be automatically generated. All the default signal plans are in NEMA phase format.

For example, in intersection node 3342, the default signal plan looks like this:



For the intersection 3346, the default phase plan looks like this:



Once you click" **Save Timing File**" and "**OK"** in the phase panel of any intersection, the input_timing.csv (default) will be saved/updated.

Section 7: Key points for attention

- Make sure the signal_representation_model in input_scenario_settings.csv are typed with the correct number. for example, "2" for **Phase Based Signal Representation Model** which we mainly discussed in the documentation.
- Make sure the demand file is correctly set in input_demand_file_list.csv.
- Before adding new phase plans in the network, change the node_type in input_node.csv to "5" indicating the pretimed signal. Make sure that all the intersection nodes are set to signalized intersection in input_node.csv.
- The timing_generator.exe could generate the input_timing_status.csv on both NEMA phase data and other phase data. However, make sure for the NEMA phase plans, all the next phase ids are set to be "-1". For others, it is the positive integer number which represents the actual next phase id.
- Make sure that the input_timing.csv and input_signal_timing_status.csv have the same number of signal intersections and same timing plans for each intersection.