Editing Road Network for SimMobility

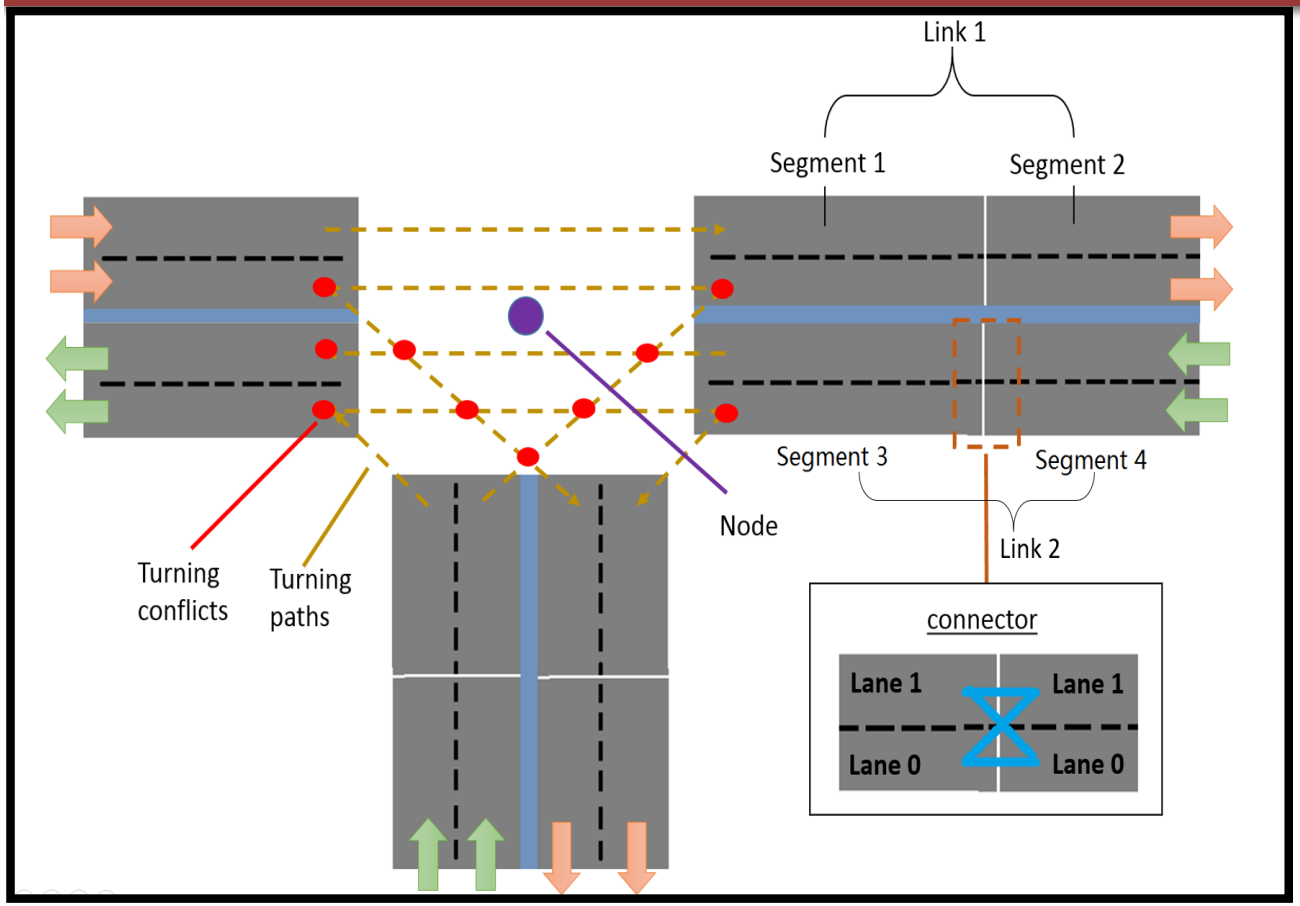
Session 7

25th Nov 2019

Agenda

- Introduction to SimMobility Road Network
- Introduction to SimMobility Public Transit Network
- Editing Road Network
- Generating Pathset
- Running simulation on edited network

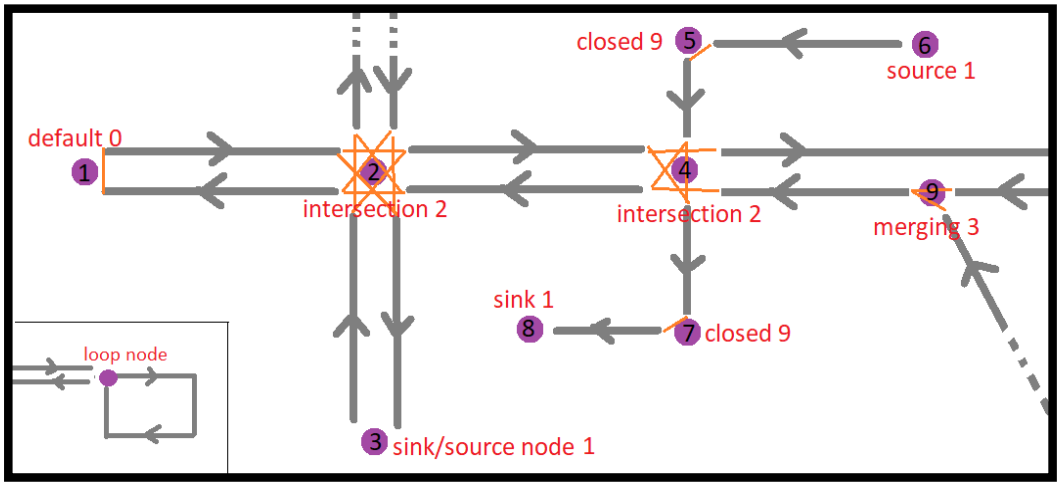
SimMobility Road Network



- Links are contiguous stretches of single directional roads.
- Nodes are the end points of links.
- Segments are sub-divisions of a link. These are typically based on changes in the link attribute (#lanes, speed-limit).
- Lanes are a part of segments that are designated for use by a single line of vehicles.
- Connectors connect lanes of two consecutive segments within the same link.
- Turning path is a path connecting specific lanes of two connected links.

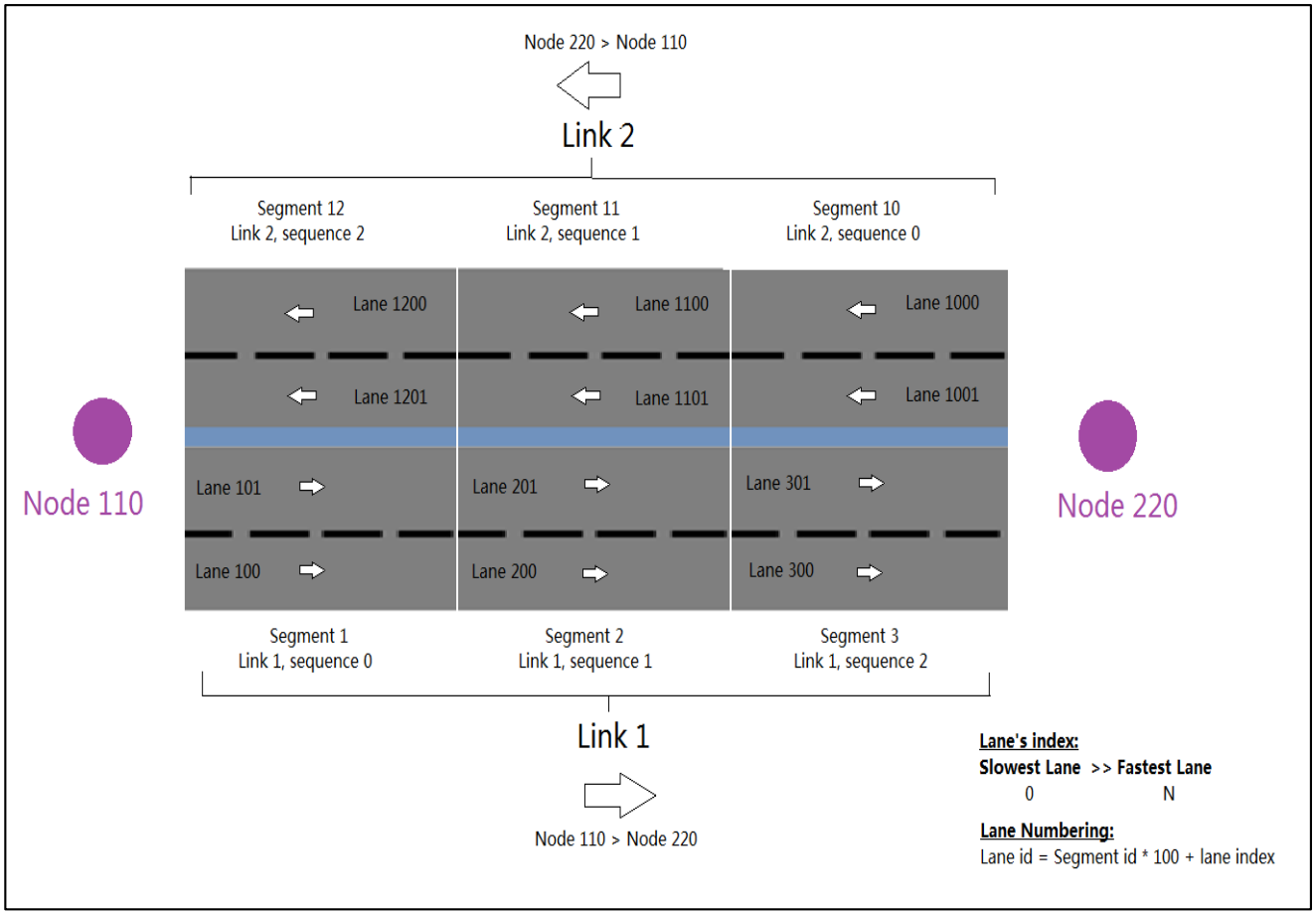
LinkCat	category id	road type	road type description	lane width
A	1	1	Expressway	3.6
B	2	2	Urban	3.5
C	3	2	Urban	3.2
D	4	5	Access	3
E	5	5	Access	2.8
SLIPROAD	6	3	RAMP	3
ROUNDABOUT	7	4	ROUNDABOUT	3

Type of Nodes in SimMobility



Node Type	Description	Comments
0	Default Node	
1	sink/source node. Source node don't have outstream link. Sink node don't have downstream link.	*please avoid demand at this type of node.*
2	intersection node	
3	merging/diverging node	
9	unreachable / closed nodes	*please avoid demand at this type of node.*
8	Loop node	*please avoid demand at this type of node.*

Lanes



max_speed. Obtained the speed limit from One Motoring website (https://www.onemotoring.com.sg/content/onemotoring/home/driving/road_safety_and_vehicle_rules/driving-rules.html#Check_speed_limit)

Lanes are a part of segments that are designated for use by a single line of vehicles.

Lane's id = segment *100 + lane's index

Connectors

Nlane to Nlane	connector
1-1, 1-2, …, 1-N, N-1	Only true connectors.
2-2	2 true connectors (green), 2 false connectors (red).
2-3	
2-4	
2-5	
3-2	
3-3	

3-4	
3-5	
3-6	
4-2	
4-3	
4-4	
4-5	

is_true_connector
1
0

Nlane to Nlane	connector
1-1, 1-2, …, 1-N, N-1	Only true connectors.
2-2	2 true connectors (green), 2 false connectors (red).
2-3	
2-4	
2-5	
3-2	
3-3	

3-4	
3-5	
3-6	
4-2	
4-3	
4-4	
4-5	

is_true_connector
1
0

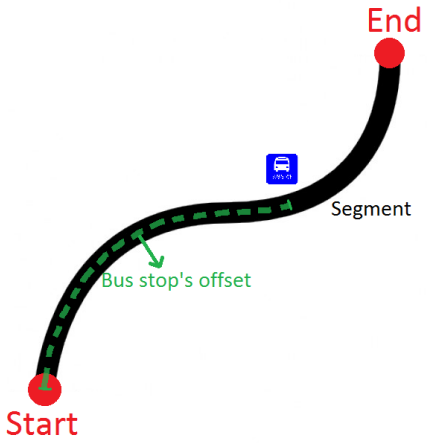
Connector is the connection between upstream and downstream segment.

True connector 1, physically connected between two segment (green lines).

True connector 0, even though they are not connected physically, but the cars allowed to change lane by using these connector (orange), especially needed in short term.

One upstream lane, only allowed to have maximum 3 connectors (true connector 1 or 0).

Bus Network in SimMobility (1/3)



- bus stop.
  - id. Random bus stop id defined by SimMobility.
  - code. Bus stop code used in the real world.
    - Used in pt bus stop, PT graph (pt_vertex and pt_edge).
    - (Singapore) Most of the bus terminal's code end with "009"
  - status.
    - OP: operation
    - NOP: non-operation (no bus serving)
    - V: Virtual Bus Stop, which created for correcting bus route.
  - terminal. 0: Not bus terminal, 1: Bus Terminal
  - length. Length of bus stop bay. (Checked manually from Google Street View)
  - section_offset. Segment offset, the distance from the starting of the segment to bus stop along the segment line.

- pt_bus_stop. Bus stop sequence of each bus line
  - Sequence_no starting with 0.
  - Obtained from GTFS 2011 and change to SimMobility format.
  - Added Virtual Bus Stop in the list for correcting bus route.
- pt_bus_dispatch_freq. Bus frequency
  - Obtained from GTFS 2011.

Bus Network in SimMobility (2/3)

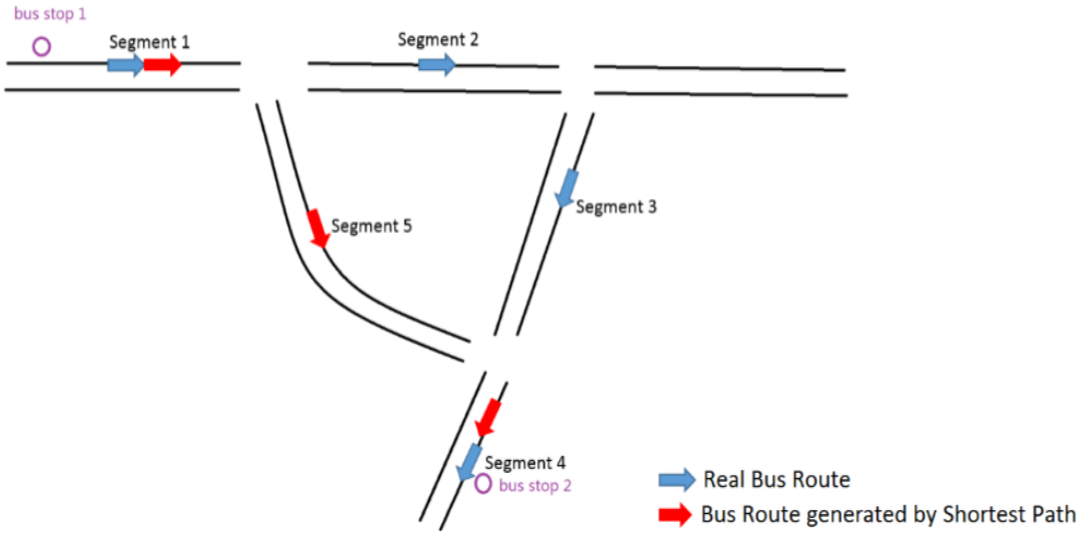


table: bus_stop

x	y	z	id	code	section_id	name	status	terminal	length	section_offset	tags	reverse_section	terminal node
..	..	0	1	1	1	BLK 457	OP	0	15	155	NULL	0	0
..	..	0	2	2	4	OPP BLK 352	OP	0	15	104	NULL	0	0

table: pt_bus_stops

route_id	stop_code	sequence_no
100_1	1	0
100_1	2	1

table: pt_bus_routes (result)

route_id	sequence_no	section_id
100_1	0	1
100_1	1	5
100_1	2	4

- pt_bus_route. Bus route (segment sequence) for each bus line.
  - Generated by SimMobility MT supply/withinday, by input bus stop, pt bus dispatch frequency and pt bus stop. From the pt bus stop, SimMobility find the shortest path between bus stops. But, the bus route result have to verified.
    The bus route can be corrected by adding the virtual bus stop at the segment, then add the virtual bus stop in the pt bus stop list, and re-generate bus route.

Bus Network in SimMobility (3/3)

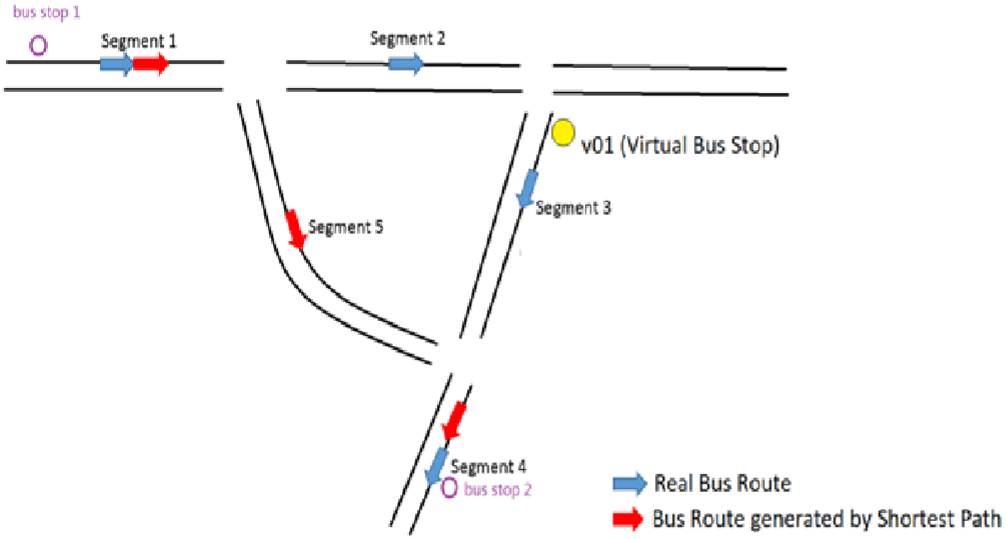


table: bus_stop

x	y	z	id	code	section_id	name	status	terminal	length	section_offset	tags	reverse_section	terminal node
..	..	0	1	1	1	BLK 457	OP	0	15	155	NULL	0	0
..	..	0	2	2	4	OPP BLK 352	OP	0	15	104	NULL	0	0
..	..	0	99901	v01	3	Virtual Bus Stop	V	0	0	150	NULL	0	0

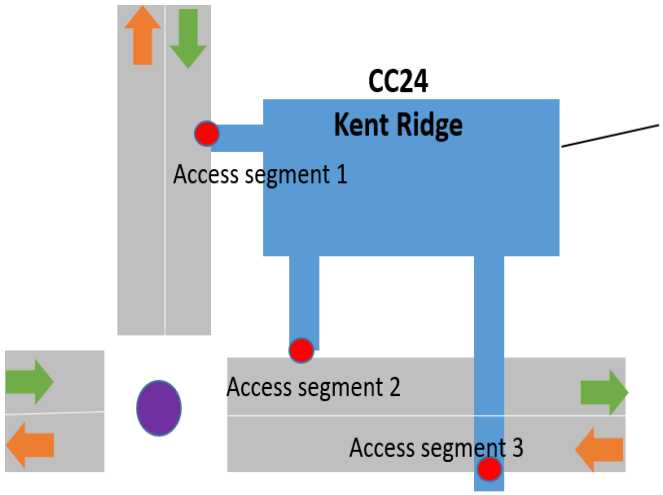
table: pt_bus_stops

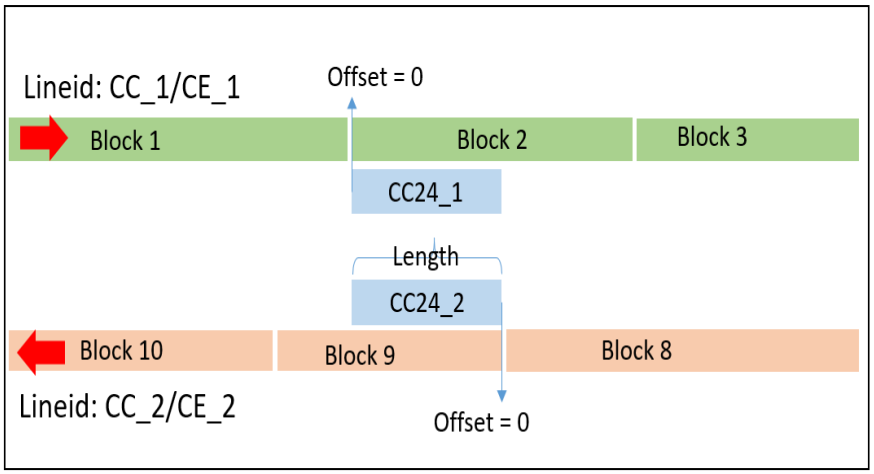
route_id	stop_code	sequence_no
100_1	1	0
100_1	2	1
100_1	v01	2

table: pt_bus_routes (result)

route_id	sequence_no	section_id
100_1	0	1
100_1	1	2
100_1	2	3
100_1	3	4

Train Network in SimMobility (1/3)





pt_train_route_platform

Line_id	platform_no	Sequence_no
CC_1	...	...
CC_1	CC23_1	5
CC_1	CC24_1	6
CC_1	CC25_1	7
CC_1	...	...

pt_train_route

Line_id	Block_id	Sequence_no
CC_1	...	...
CC_1	1	5
CC_1	2	6
CC_1	3	7
CC_1	...	...

- train stop. Train station.
  - Id. Random id by SimMobility
  - platform_name. E.g.CC24 (for single station), CC22/EW21 (interchange station)
  - station_name. E.g. Kent Ridge, Buona Vista
  - type. (For Singapore case, MRT and LRT)
  - op_year. Operation year. E.g. 2015, 2015/2012 (yr 2015 for CC22, yr 2012 for EW21)

Train Network in SimMobility (2/3)

- train access segment
  - the access segment to train station.
- pt train dispatch freq
  - Obtained from GTFS 2011, SMRT website (for Singapore case)
  - End_time. End before time 23:59.
- pt block & pt block polyline
  - train track with single directional.
  - **No multiple train platforms sharing same block. Keep it one-to-one relationship (pt block – train platform)
- train platform. Platforms in the train station which served single directional train service line.
  - platform_no. E.g. CC24_1
  - station_no. E.g. CC24
  - line_id. Serving train line. One platform can be served by single or multiple line. E.g. CC24_1, this platform served by two train lines (CC_1/CE_1).
  - Capacity. Platform capacity.
  - Type. 0 – single station, 1 – interchange station
  - block_id. Train track id. ** No sharing block between train platforms.
  - pos_offset. Starting of segment to the starting of the train platform, along the block
  - length. Length of train platform.

Train Network in SimMobility (3/3)

- pt train route platform
  - sequence of train platform for each train line service.
- pt train platform
  - sequence of train track route for each train line service.
- train fleet. Train fleet size.
- train line properties.
- pt opposite lines. The opposite direction train service line.
- train uturn platforms. The first train platform of train service to start the new trip.
- pt train platform transfer time. Transfer time between platforms in the train station. E.g. default 0 second for transfer to opposite train service line. And default 60 seconds for transfer between platforms with the different train line (happened at Interchange station).
- rail transit edge. The travel time between platform (to the next platform, and transfer in the station).

Taxi Stand in SimMobility

Obtained taxi stand data from LTA (LTA Datamall).

- taxi stand.
  - segment_id. Nearest segment id
  - length. Segment's length
  - segment_offset. The distance from the starting of segment to the taxi stand along the segment line.
  - tags. There are few type of taxi stand which mentioned in this attribute, they are "TAXI STAND", "TAXI STOP", ""TAXI PICK UP / DROP OFF", "AMOD PUDO" (for Amod case).

Editing Road Network in SimMobility

Add/Edit
- Node
- Link
- Segment
- Lane
- Turning Paths

Export to CSV

Format and Merge CSV Files

Import Into Database

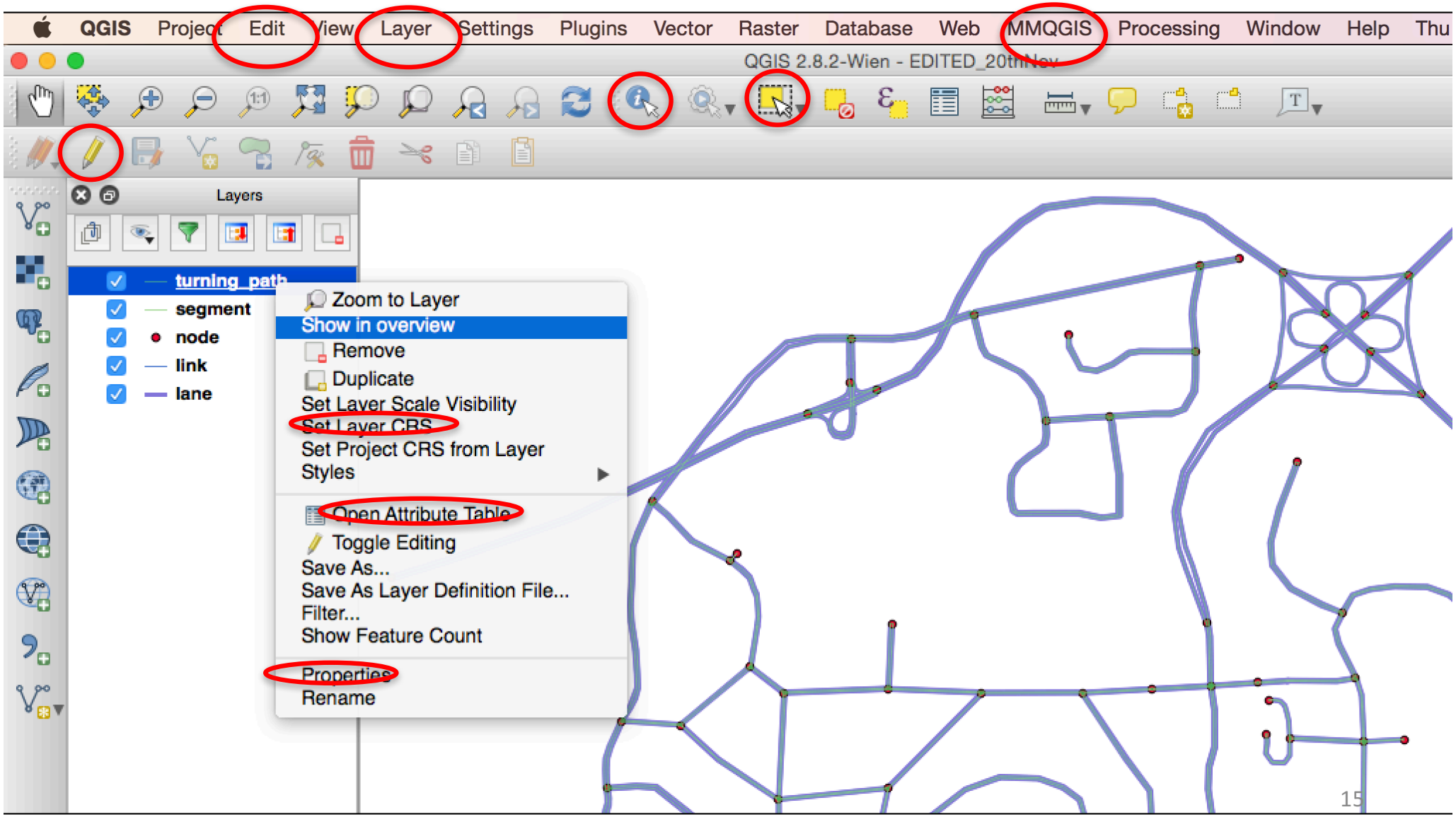
Update
- Connectors
- Link default Travel time

Validate Supply Data

Visualize Edited Network

Introduction to QGIS

QGIS is a free and open-source cross-platform desktop geographic information system application that supports viewing, editing, and analysis of geospatial data



Open SHP files

Shape files for Prototypical Cities

Go to Documents → 01_Base_Shape_File

Open QGIS

Drag and drop the following .shp files to QGIS

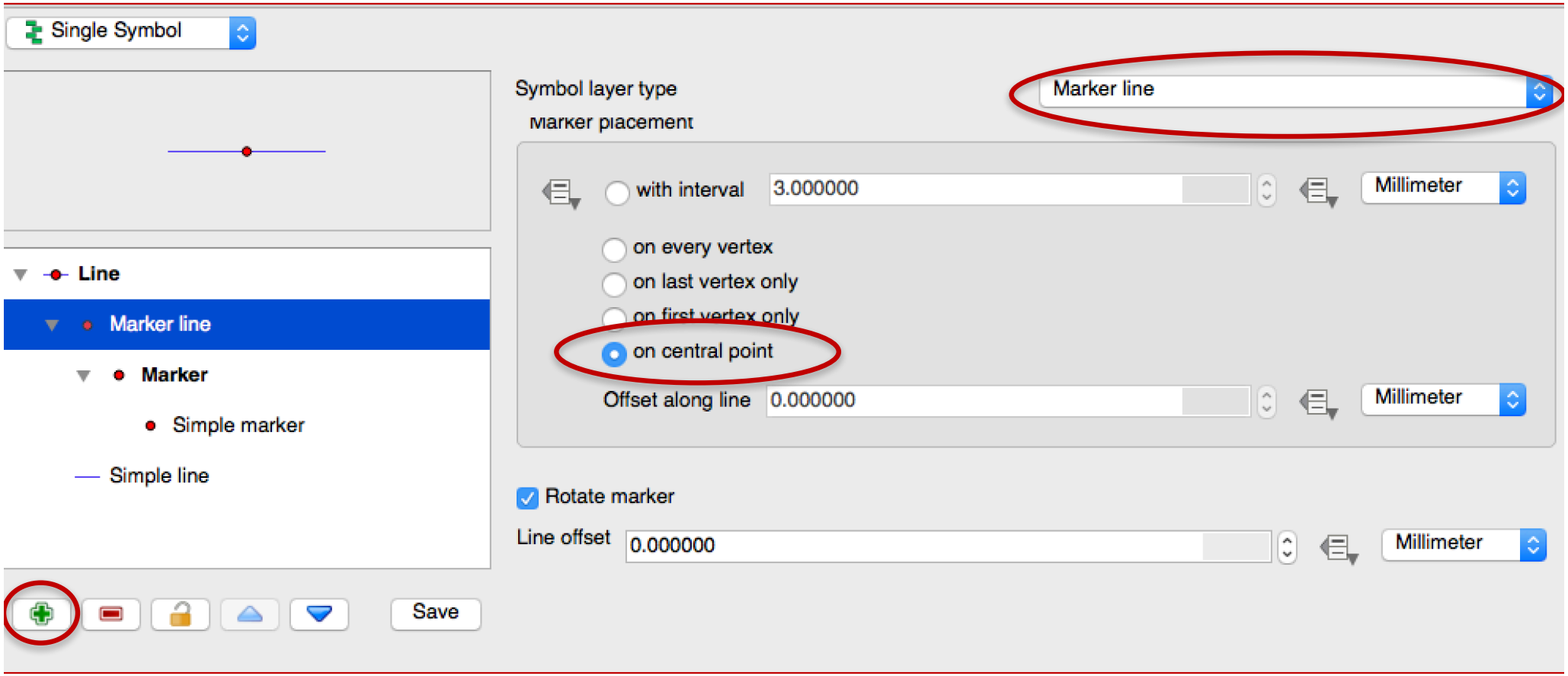
- Node
- Link
- Lane
- Segment
- Turning_Path

Turn on the direction of the line

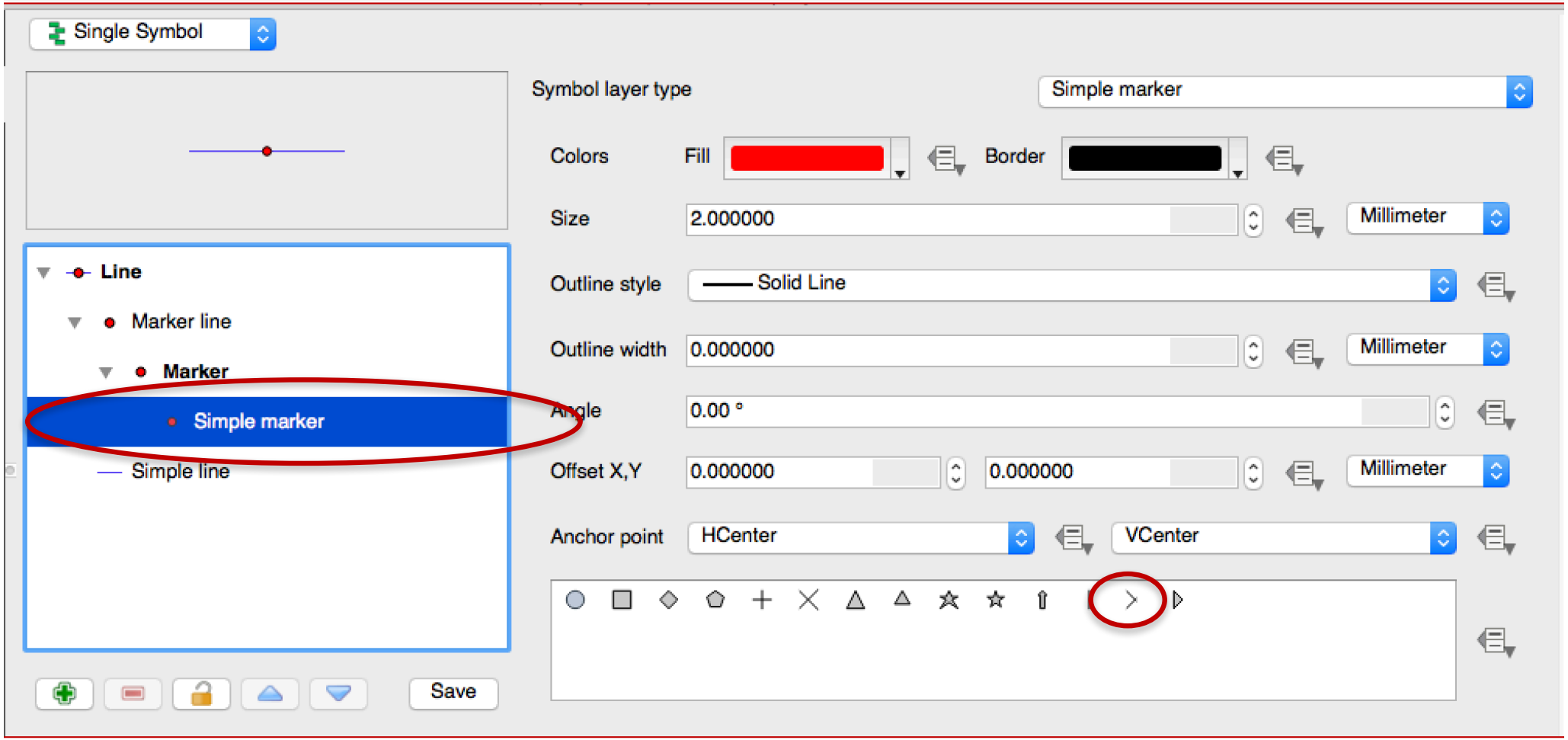
- Link
- Lane
- Turning_path

Direction of the line layers

Right click Link layer → properties

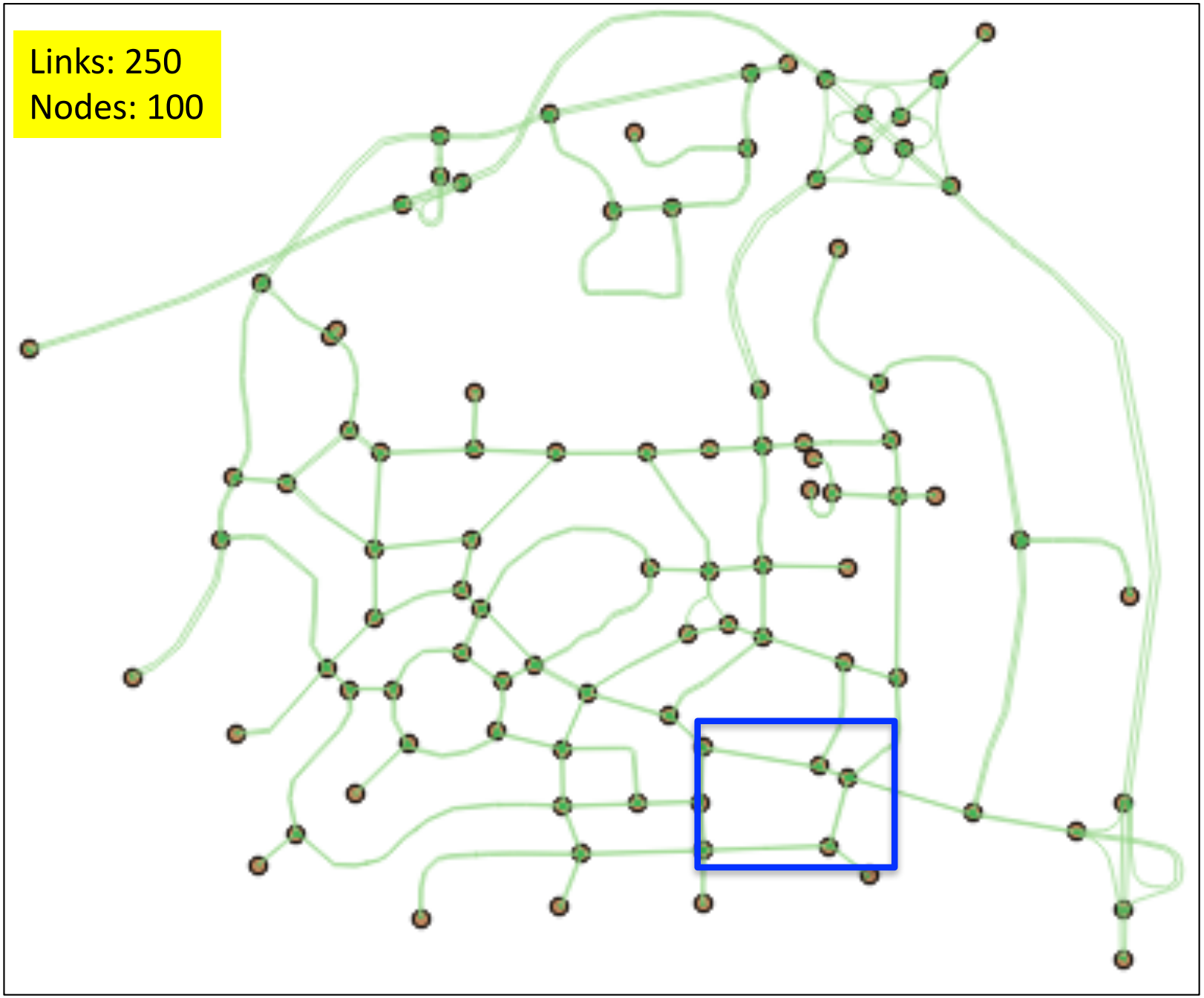


Direction of the line layers

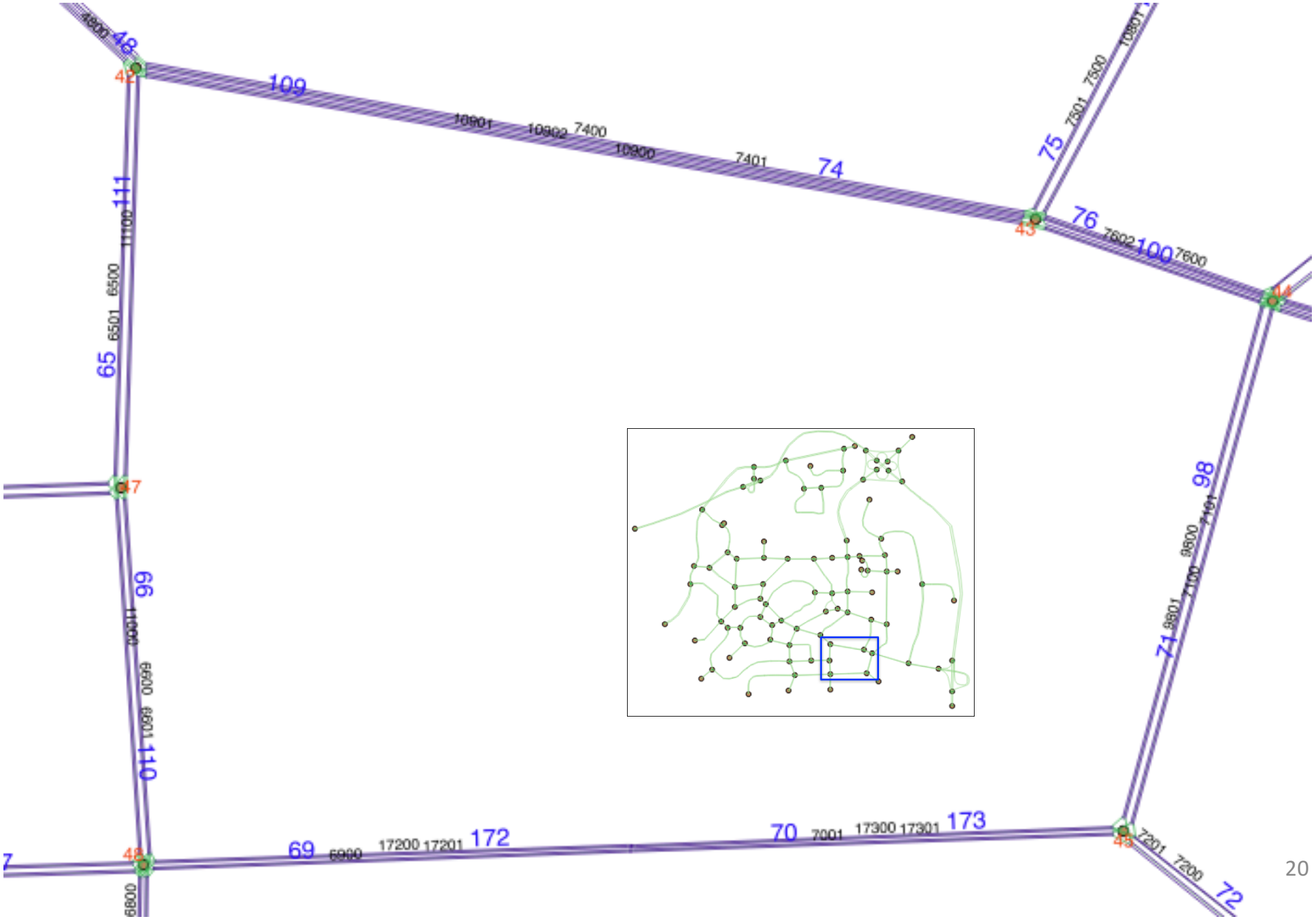


- Press OK
- Repeat the same exercise for Lane and Turning_path layers

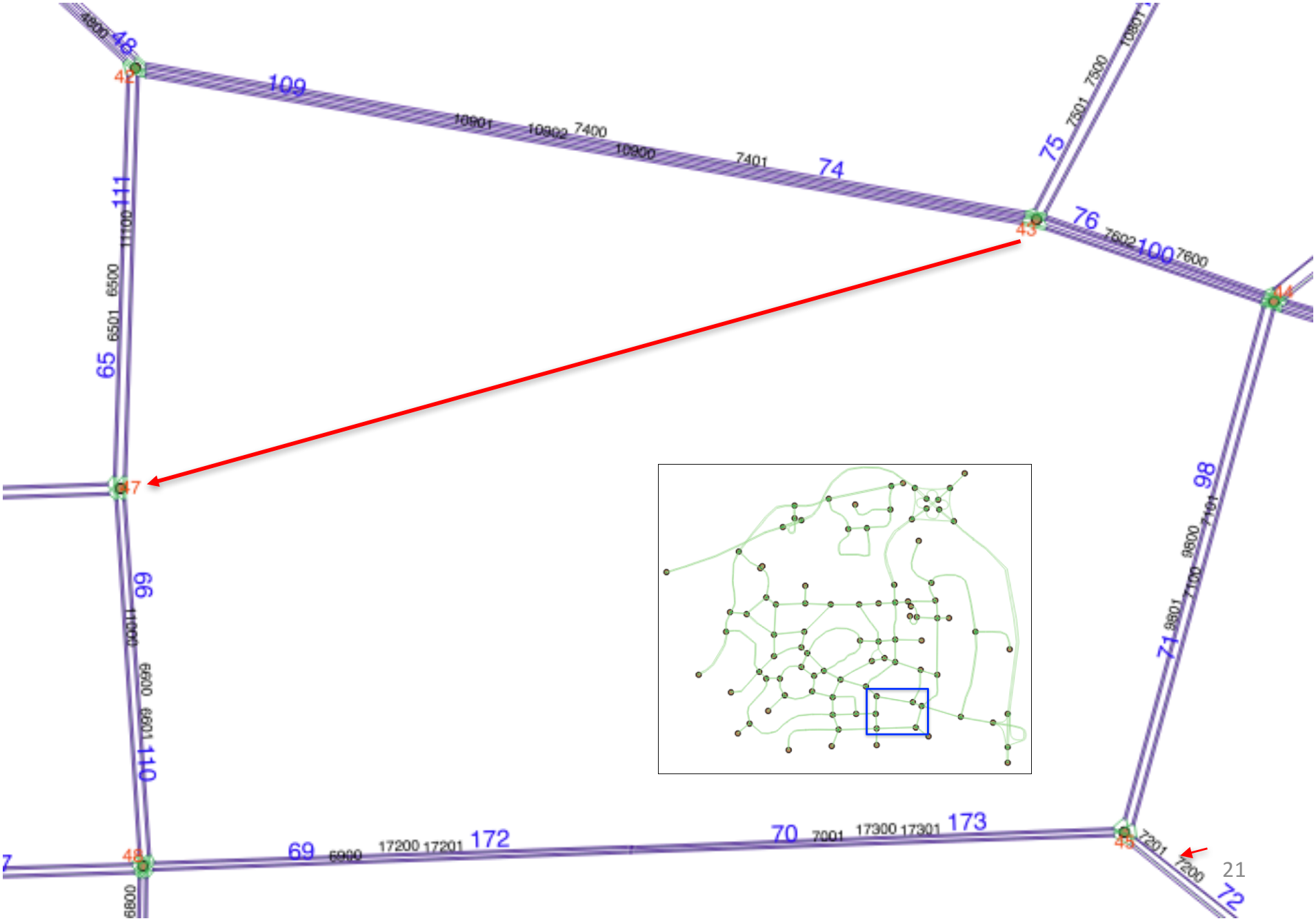
Base Network



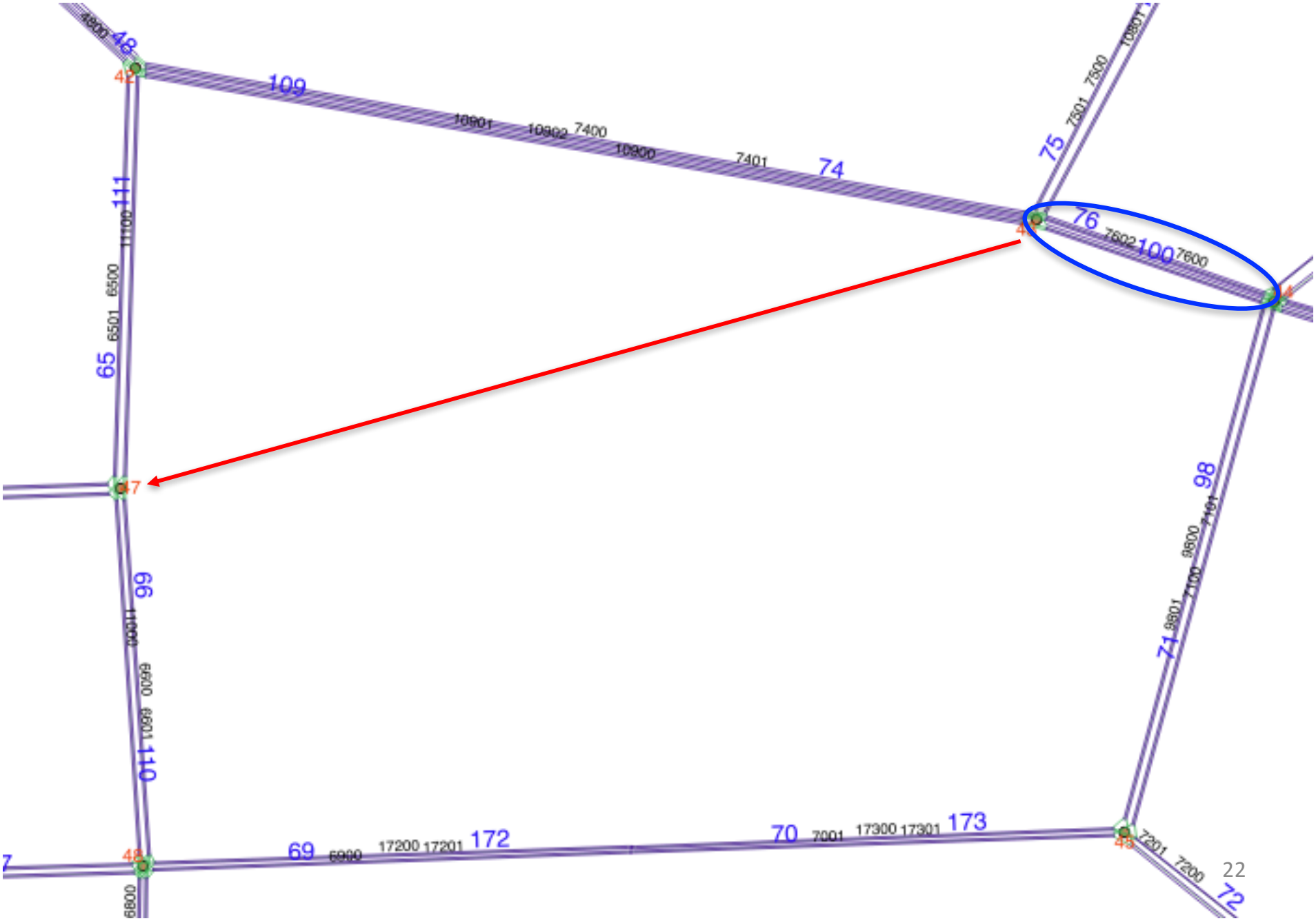
Base Network



Base Network



Base Network



Base Network

link	
road_name	Basswood Ave
(Derived)	
(clicked coordin…	371200.141763, 140534.136055
Length	316.863 m
feature id	121
firstX	371,407.4406
firstY	140,462.0583
lastX	371,107.6748
lastY	140,564.7355
(Actions)	
	View feature form
id	122
road_type	2
category	2
from_node	44
to_node	43
road_name	Basswood Ave
tags	NULL

Link

Segment

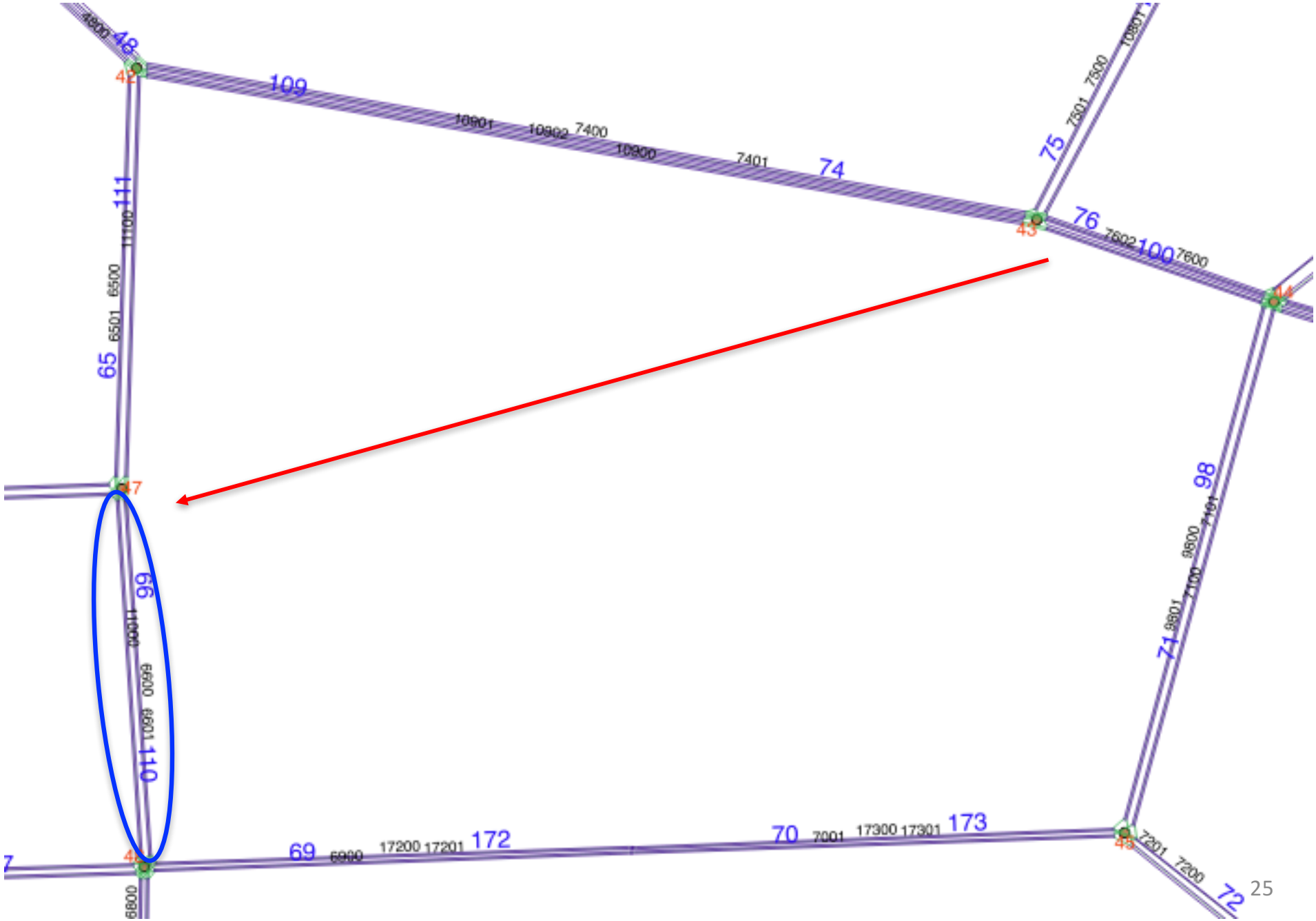
Feature	Value
segment	
id	100
(Derived)	
(clicked coordin…	371149.242447, 140551.102493
Length	316.863 m
feature id	119
firstX	371,407.4406
firstY	140,462.0583
lastX	371,107.6748
lastY	140,564.7355
(Actions)	
	View feature form
id	100
link_id	122
sequence_n	0
num_lanes	3
capacity	5700
max_speed	70
tags	<old_id>: 0,<old_fnode>: 0,<old_tnode>: 0,<…
category	2
segwidth	10.500
segwdthrad	5.250
length	316.86296
desc	NULL
linkcatspd	2_70
capperlane	1900

Base Network

Feature	Value
segment	
id	100
(Derived)	
(clicked coordin…	371243.618262, 140516.109214
Length	316.863 m
feature id	119
firstX	371,407.4406
firstY	140,462.0583
lastX	371,107.6748
lastY	140,564.7355
(Actions)	
	View feature form
id	100
link_id	122
sequence_n	0
num_lanes	3
capacity	5700
max_speed	70
tags	<old_id>: 0,<old_fnode>: 0,<old_tnode>: 0,<…
category	2
segwidth	10.500
segwdthrad	5.250
length	316.86296
desc	NULL
linkcatspd	2_70
capperlane	1900

Lane

Base Network



Base Network

Feature	Value
link	
road_name	Hemlock Street 1
(Derived)	
(clicked coordin…	369824.269633, 140129.062333
Length	496.739 m
feature id	129
firstX	369,821.1166
firstY	140,184.5114
lastX	369,855.1678
lastY	139,688.941
(Actions)	
	View feature form
id	130
road_type	2
category	3
from_node	47
to_node	48
road_name	Hemlock Street 1
tags	NULL

Link

Segment

Feature	Value
segment	
id	66
(Derived)	
(clicked coordin…	369827.45084, 140093.008651
Length	496.739 m
feature id	87
firstX	369,821.1166
firstY	140,184.5114
lastX	369,855.1678
lastY	139,688.941
(Actions)	
	View feature form
id	66
link_id	130
sequence_n	0
num_lanes	2
capacity	3400
max_speed	60
tags	<old_id>: 0,<old_fnode>: 0,<old_tnode>: 0,<…
category	3
segwidth	6.400
segwdthrad	3.200
length	496.73887
desc	NULL
linkcatspd	3_60
capperlane	1700

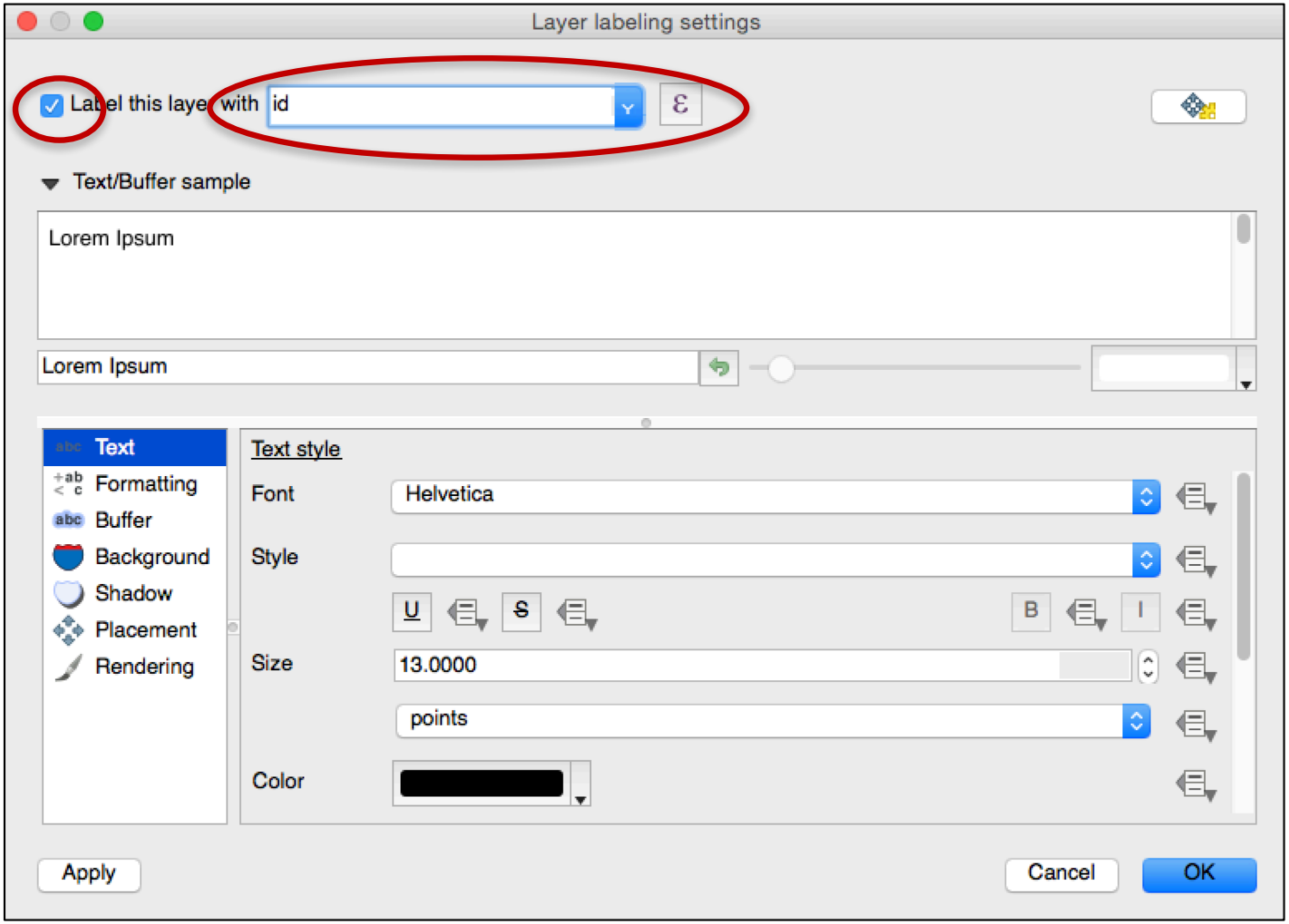
Base Network

Feature	Value
lane	
id	6600
(Derived)	
(clicked coordin…	369825.860236, 140155.042192
Length	496.739 m
feature id	110
firstX	369,822.7126
firstY	140,184.6241
lastX	369,856.7638
lastY	139,689.0537
(Actions)	
	View feature form
id	6600
width	3.2
vehicle_mo	63
bus_lane	0
can_stop	1
can_park	0
high_occ_v	0
has_road_s	0
segment_id	66
tags	NULL

Lane

Display the ID of the layers

- Select the layer → Layer → Labeling

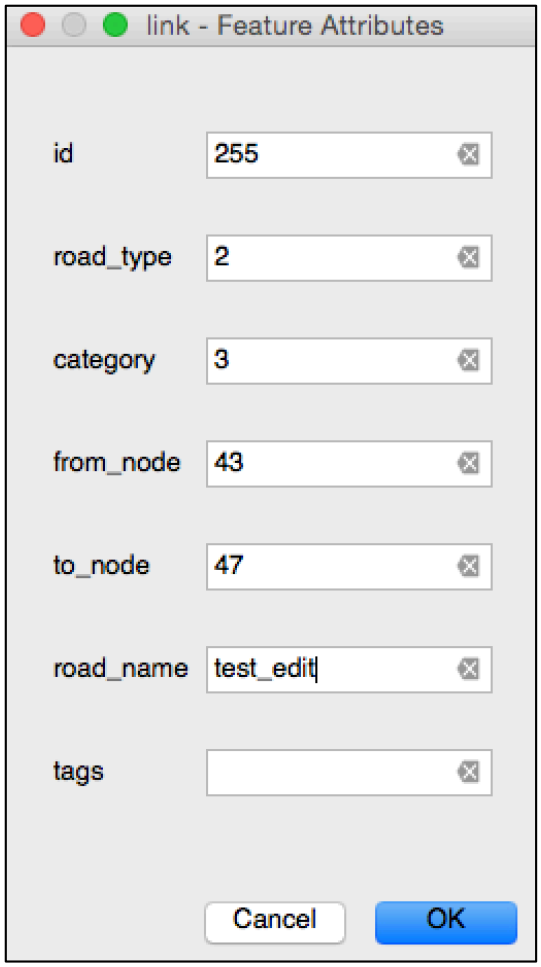


Adding a link

1. Link → Toggle Editing  → Edit → Add Feature

Double click at start position → drag to the end → right click → Fill in the attributes → save 

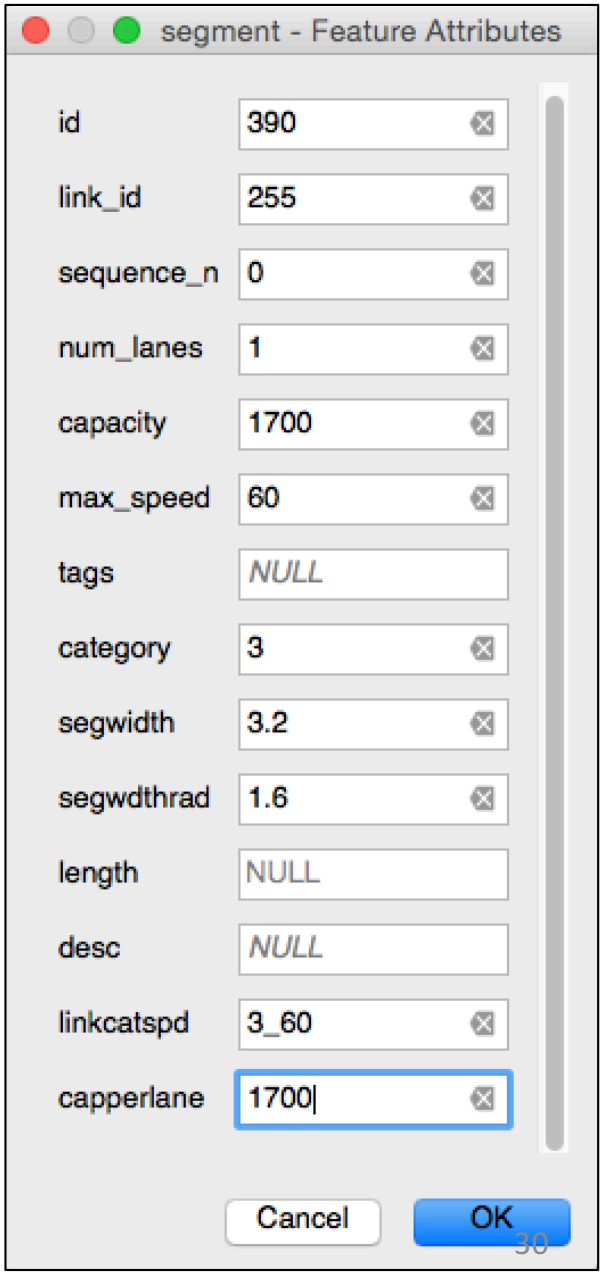
(Leave some space from the node to start and end the link)



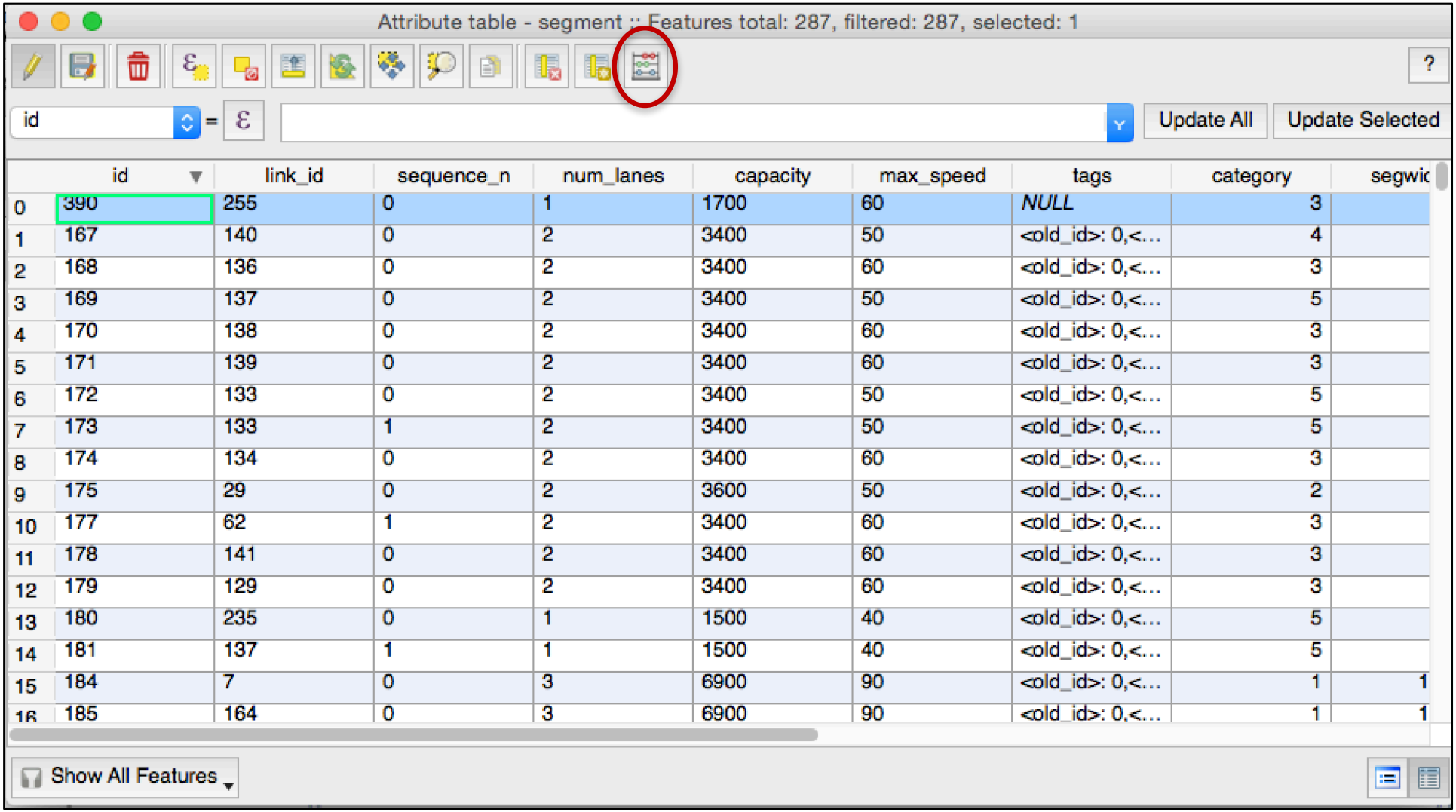
category id	road type	road type description
1	1	Expressway
2	2	Urban
3	2	Urban
4	5	Access
5	5	Access
6	3	RAMP
7	4	ROUNDABOUT

Adding a segment

1. Segment → Toggle Editing 
2. Select newly created link → crtl-C
3. Select the segment layer → crtl-V
4. Go to  → click the newly created segment
5. Go to ”Edit feature from” → input the attributes → 

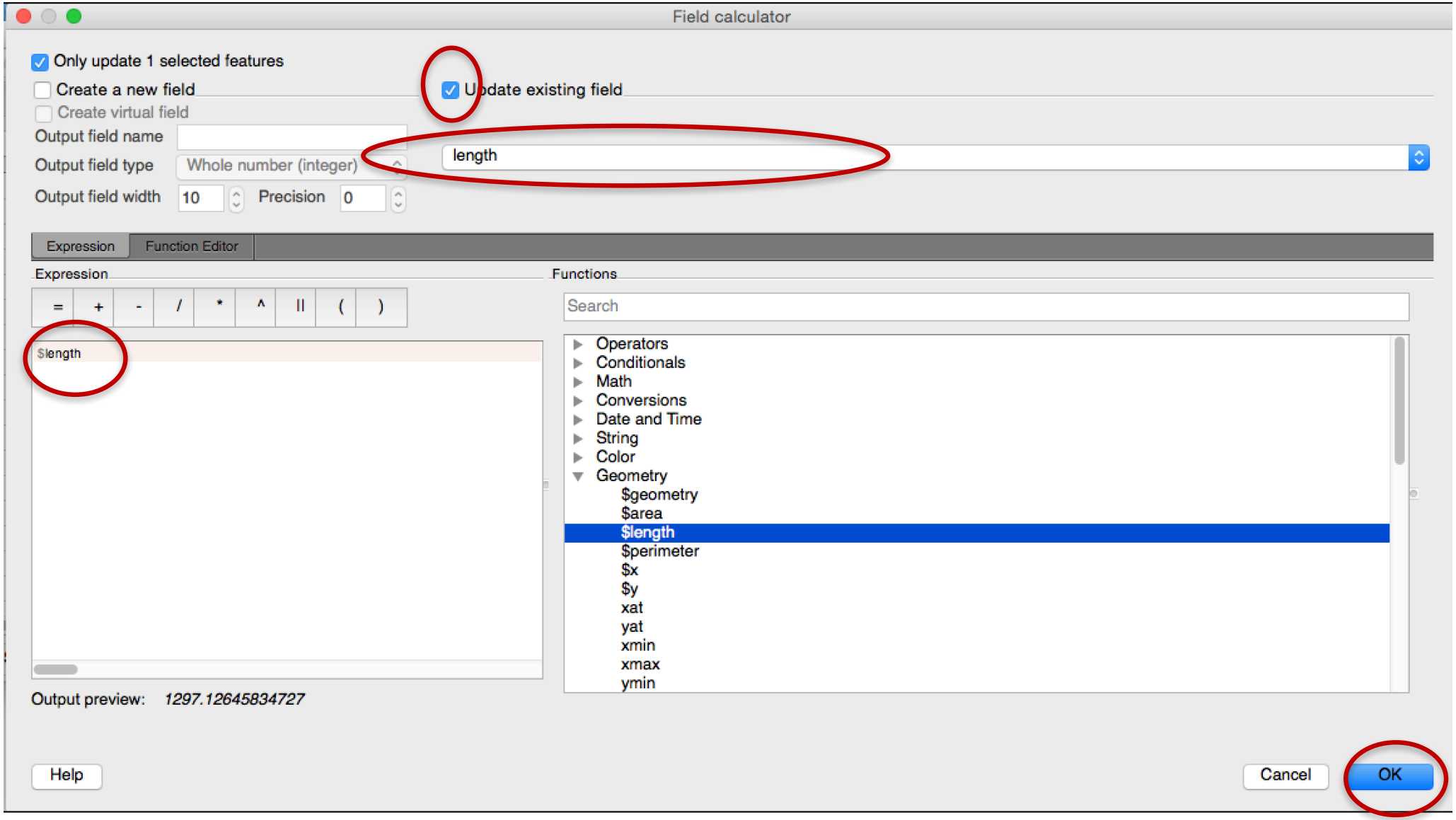


Adding a segment length



	id	link_id	sequence_n	num_lanes	capacity	max_speed	tags	category	segwic
0	390	255	0	1	1700	60	*NULL*	3	
1	167	140	0	2	3400	50	<old_id>: 0,<...	4	
2	168	136	0	2	3400	60	<old_id>: 0,<...	3	
3	169	137	0	2	3400	50	<old_id>: 0,<...	5	
4	170	138	0	2	3400	60	<old_id>: 0,<...	3	
5	171	139	0	2	3400	60	<old_id>: 0,<...	3	
6	172	133	0	2	3400	50	<old_id>: 0,<...	5	
7	173	133	1	2	3400	50	<old_id>: 0,<...	5	
8	174	134	0	2	3400	60	<old_id>: 0,<...	3	
9	175	29	0	2	3600	50	<old_id>: 0,<...	2	
10	177	62	1	2	3400	60	<old_id>: 0,<...	3	
11	178	141	0	2	3400	60	<old_id>: 0,<...	3	
12	179	129	0	2	3400	60	<old_id>: 0,<...	3	
13	180	235	0	1	1500	40	<old_id>: 0,<...	5	
14	181	137	1	1	1500	40	<old_id>: 0,<...	5	
15	184	7	0	3	6900	90	<old_id>: 0,<...	1	1
16	185	164	0	3	6900	90	<old_id>: 0,<...	1	1

Adding a segment length



Adding a segment length

Open attribute table if all the values are ok save the changes  

Attribute table - segment :: Features total: 287, filtered: 287, selected: 1

id = ε | Update All | Update Selected

	id	link_id	sequence_n	num_lanes	capacity	max_speed	tags	category	segwidth	segwdthrad	length	
0	390	255	0	1	1700	60	NULL	3	3.200	1.600	1297.12646	NU
1	167	140	0	2	3400	50	<old_id>: 0,<...	4	6.000	3.000	674.03179	NU
2	168	136	0	2	3400	60	<old_id>: 0,<...	3	6.400	3.200	524.29779	NU
3	169	137	0	2	3400	50	<old_id>: 0,<...	5	5.600	2.800	1113.94464	NU
4	170	138	0	2	3400	60	<old_id>: 0,<...	3	6.400	3.200	588.27858	NU
5	171	139	0	2	3400	60	<old_id>: 0,<...	3	6.400	3.200	1301.67514	NU
6	172	133	0	2	3400	50	<old_id>: 0,<...	5	2.800	1.400	666.26253	NU
7	173	133	1	2	3400	50	<old_id>: 0,<...	5	2.800	1.400	672.38958	NU
8	174	134	0	2	3400	60	<old_id>: 0,<...	3	6.400	3.200	567.90383	NU
9	175	29	0	2	3600	50	<old_id>: 0,<...	2	3.500	1.750	529.89530	chg
10	177	62	1	2	3400	60	<old_id>: 0,<...	3	6.400	3.200	265.38880	NU
11	178	141	0	2	3400	60	<old_id>: 0,<...	3	6.400	3.200	768.57885	NU
12	179	129	0	2	3400	60	<old_id>: 0,<...	3	6.400	3.200	647.88223	NU
13	180	235	0	1	1500	40	<old_id>: 0,<...	5	2.800	1.400	1129.55318	NU
14	181	137	1	1	1500	40	<old_id>: 0,<...	5	2.800	1.400	1098.23802	NU
15	184	7	0	3	6900	90	<old_id>: 0,<...	1	10.800	5.400	580.48107	NU
16	185	164	0	3	6900	90	<old_id>: 0,<...	1	10.800	5.400	578.83604	NU

Show All Features

Adding a lane

1. Segment → Toggle Editing 

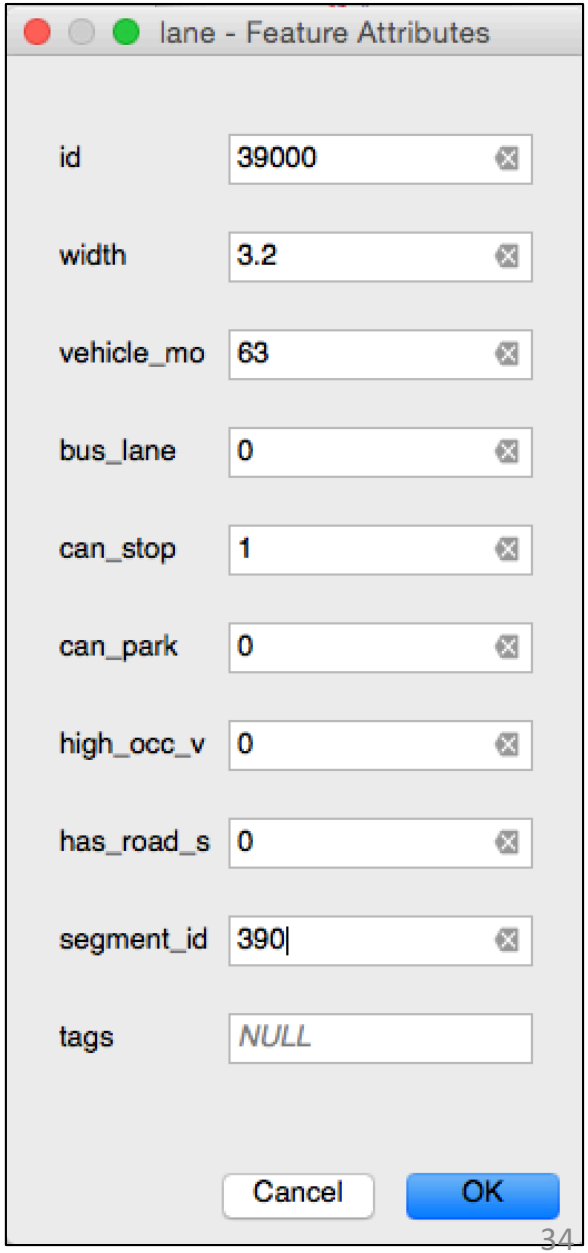
2. Select newly created segment→ crtl-C

3. Select the lanelayer → crtl-V → save 

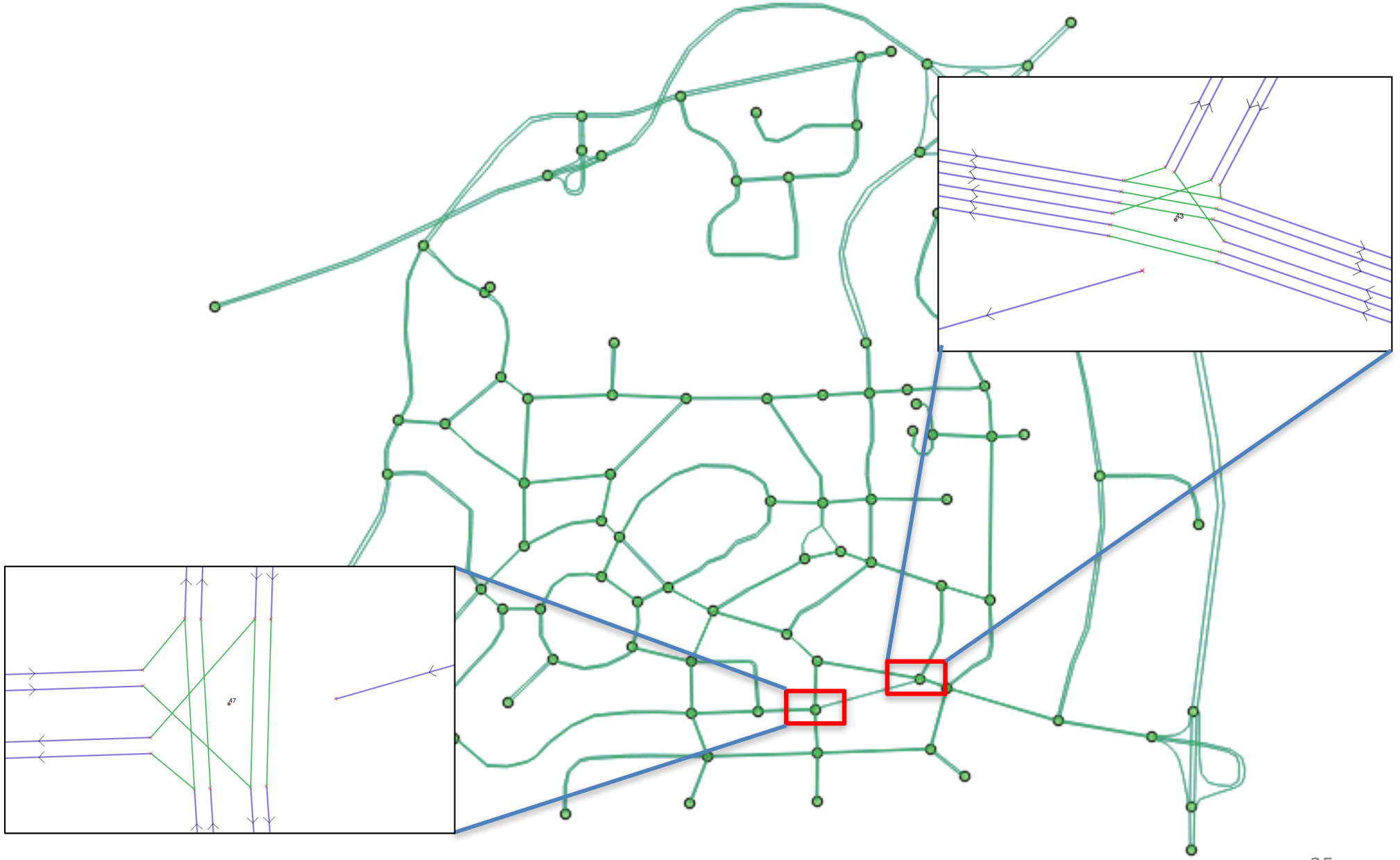
4. Go to  → click the newly created segment

5. Go to ”Edit feature from” → input the attributes →

description	roadtype	linkcategory	lanewidth
Expressway	1	1	3.6
Urban	2	2	3.5
Urban	2	3	3.2
Access	5	4	3
Access	5	5	2.8
Sliproad	3	6	3
Roundabout	4	7	3

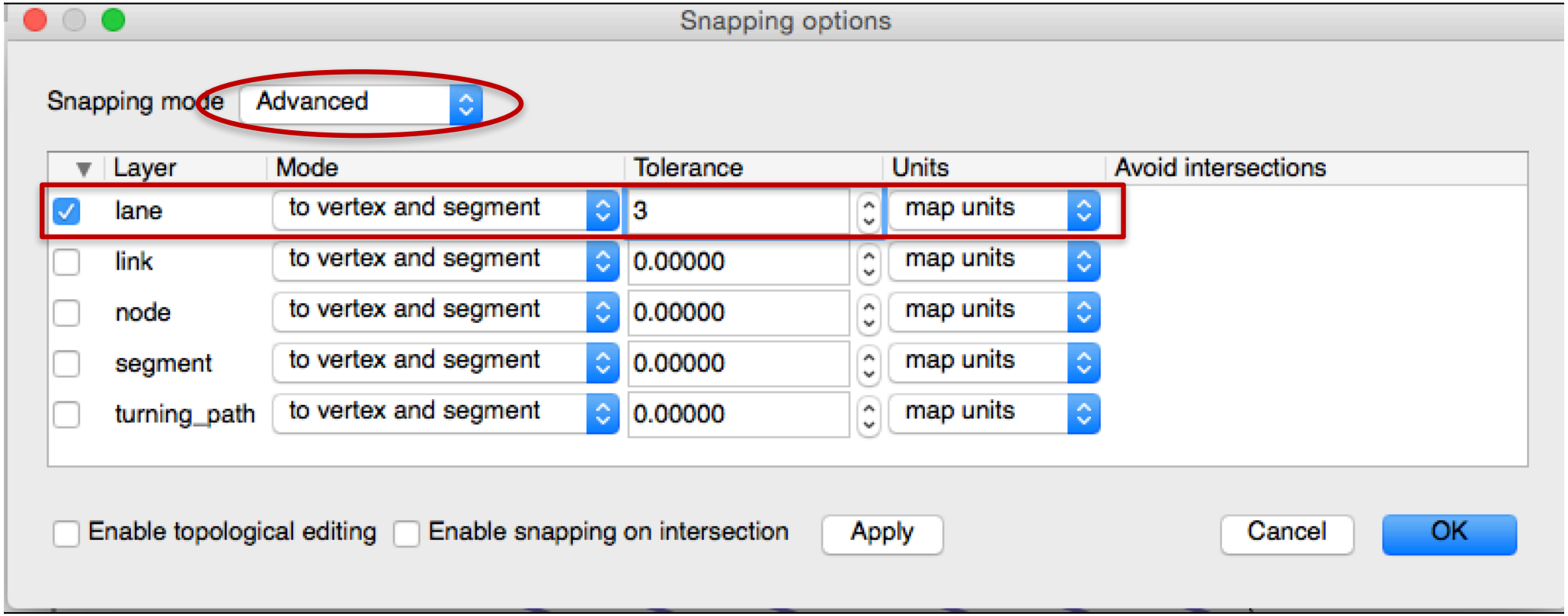


Updated Network



Adding Turning Paths

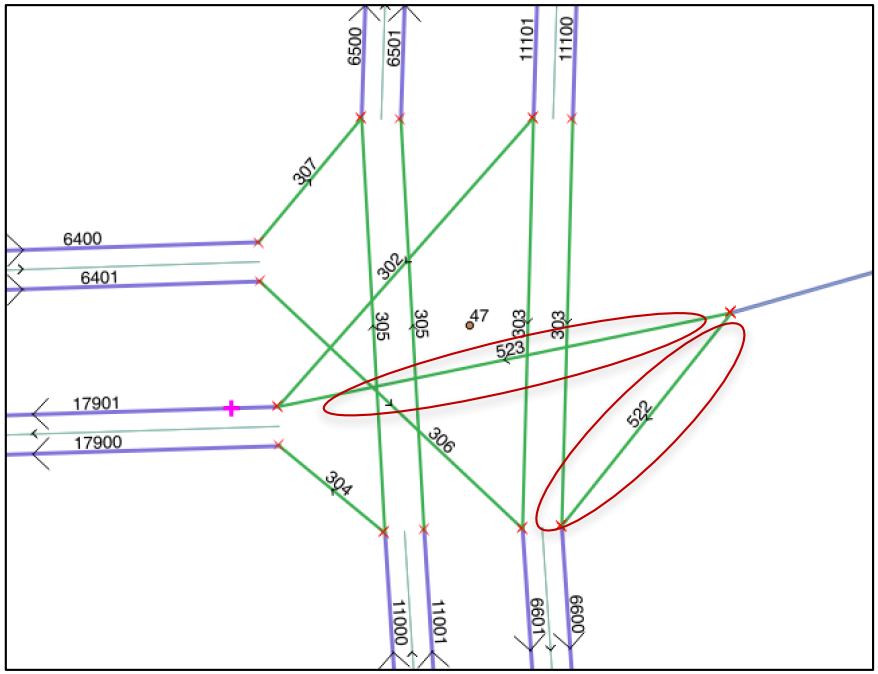
Enable Snapping: Settings → Digitizing → Snapping

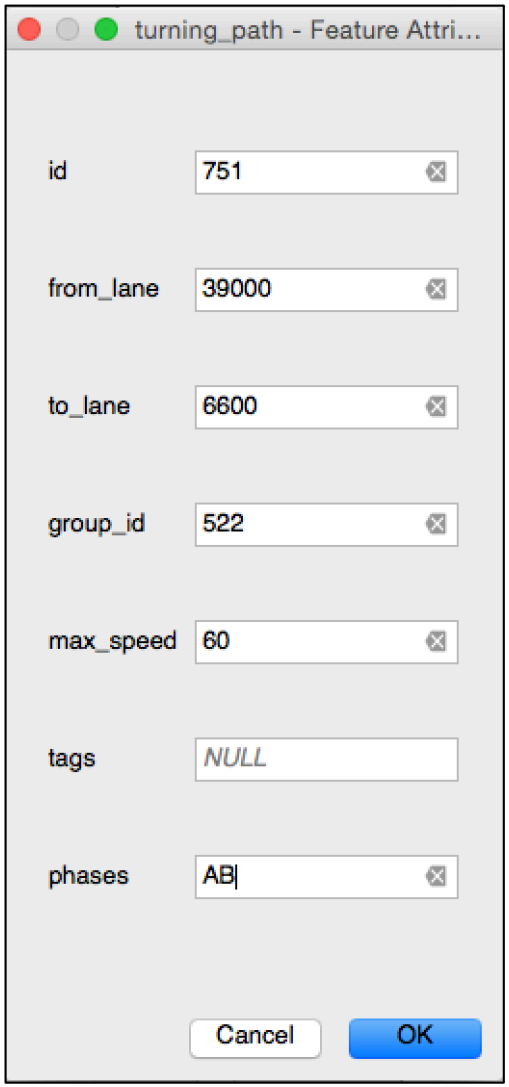


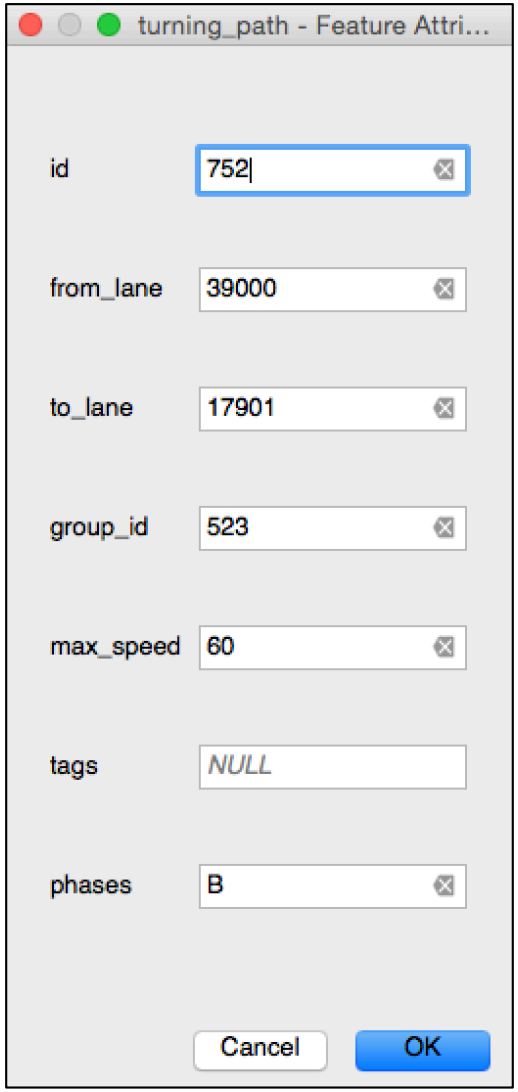
1. Turning Path → Toggle Editing → Edit → Add Feature → Drag and drop → fill in the attributes

Adding Turning Paths

1. Turning Path → Toggle Editing → Edit → Add Feature → Drag and drop → fill in the attributes

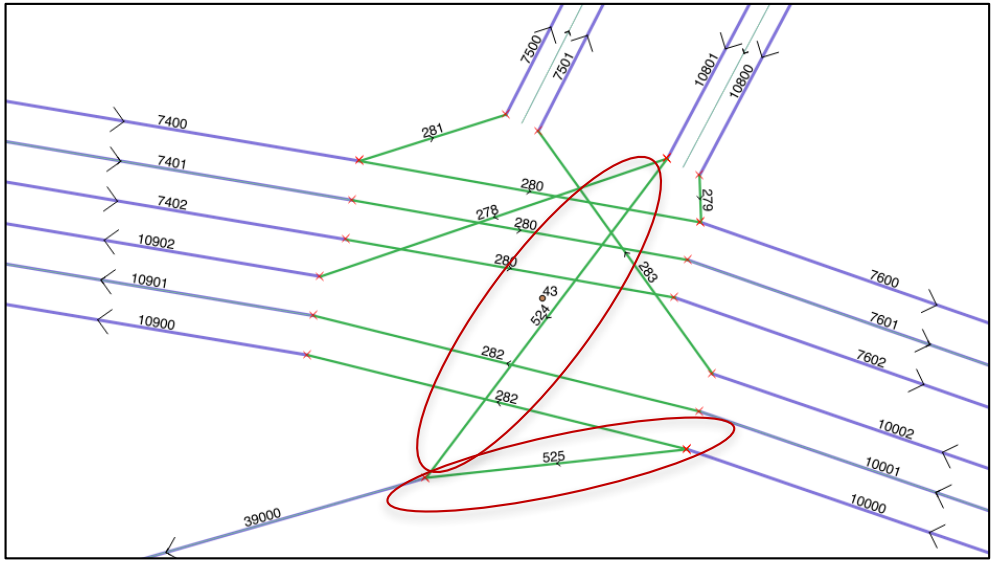


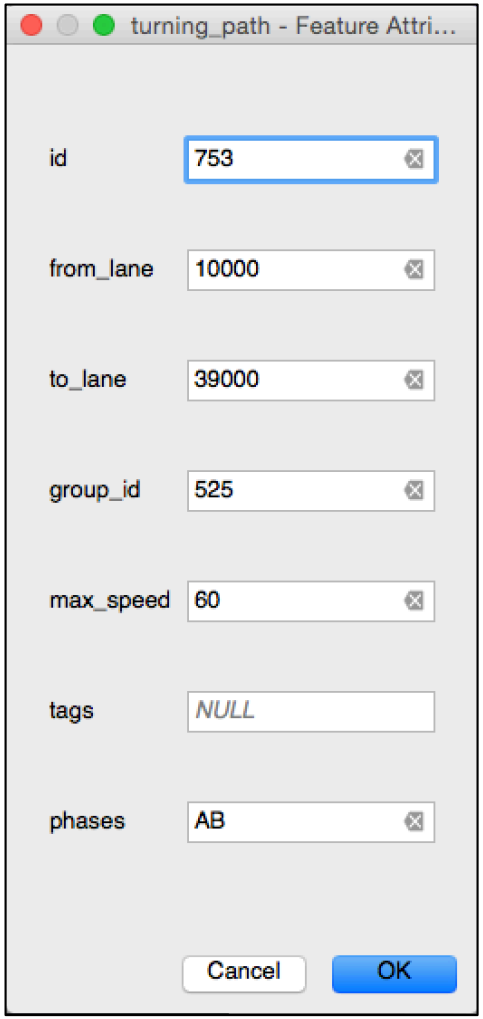


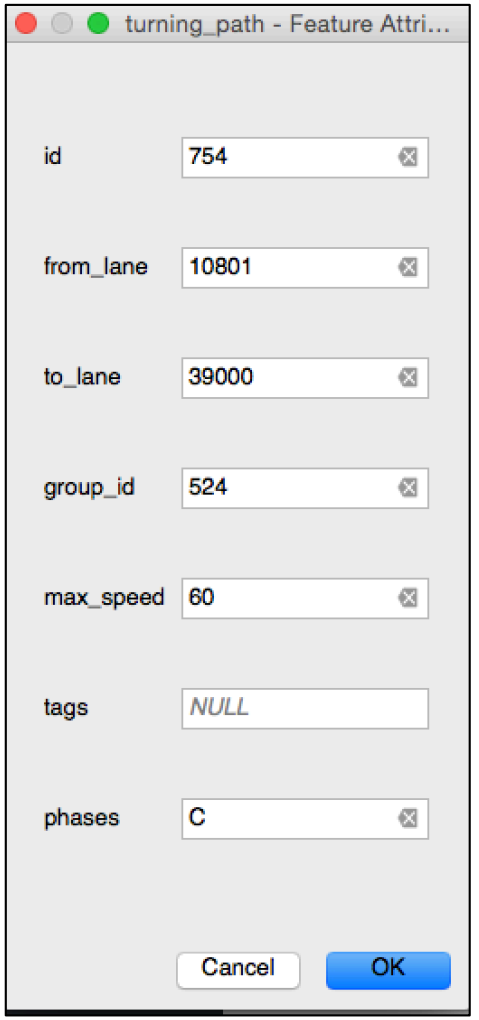


Adding Turning Paths

1. Turning Path → Toggle Editing → Edit → Add Feature → Drag and drop → fill in the attributes

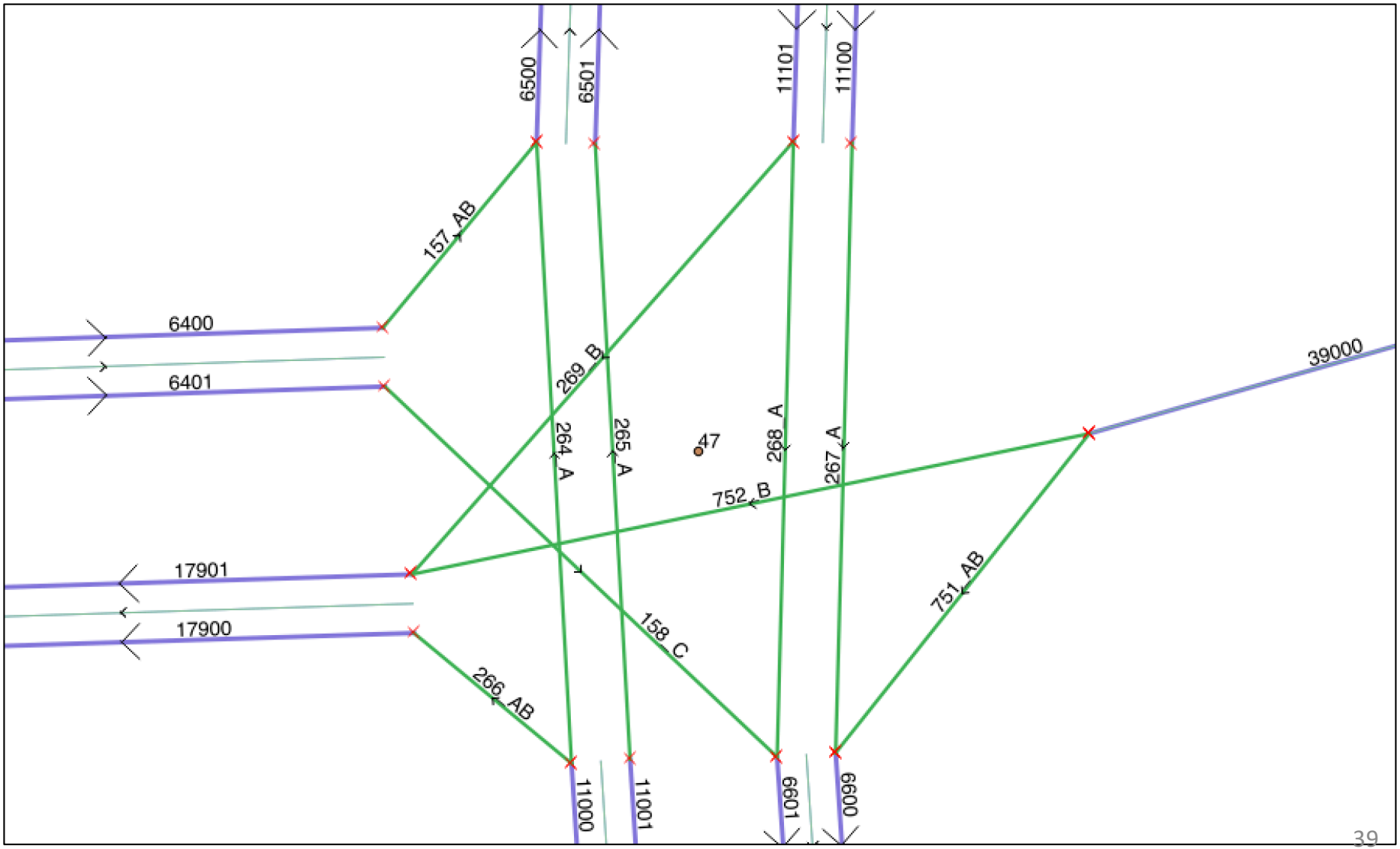






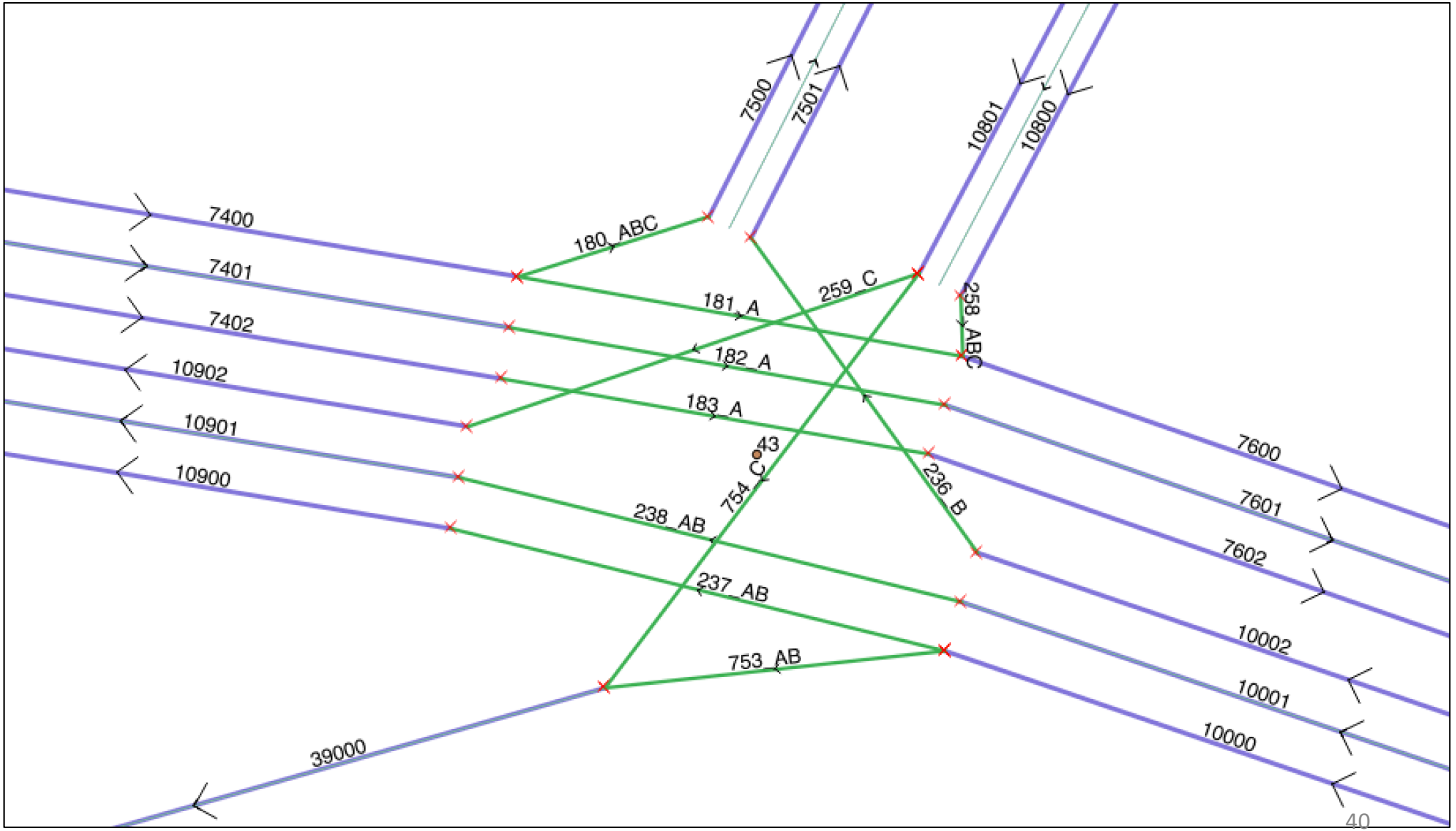
Adding Phases

Making an un-signalised intersection to signalised



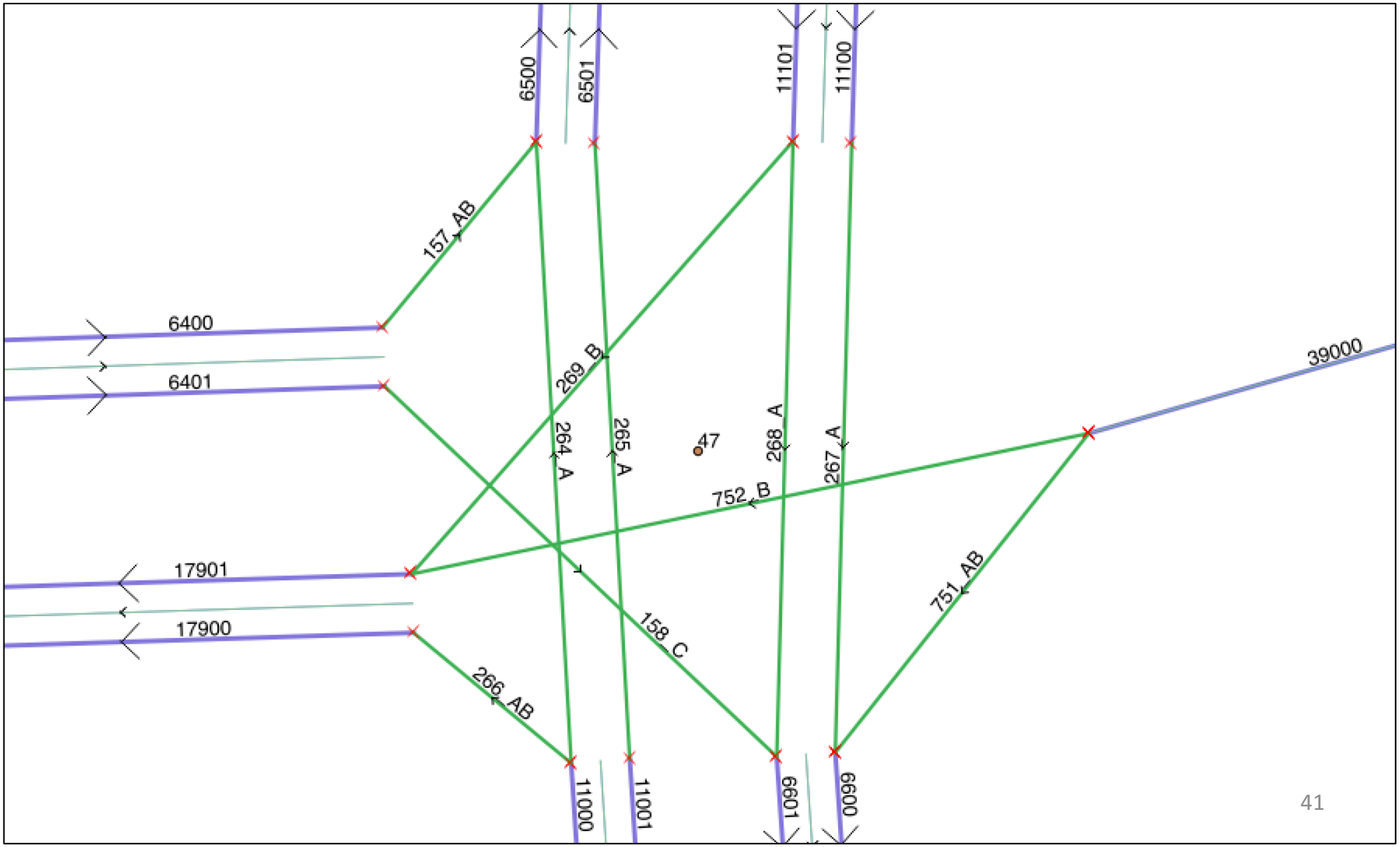
Check the newly added layers

Node 43



Check the newly added layers

Node 47



Save the shape layers

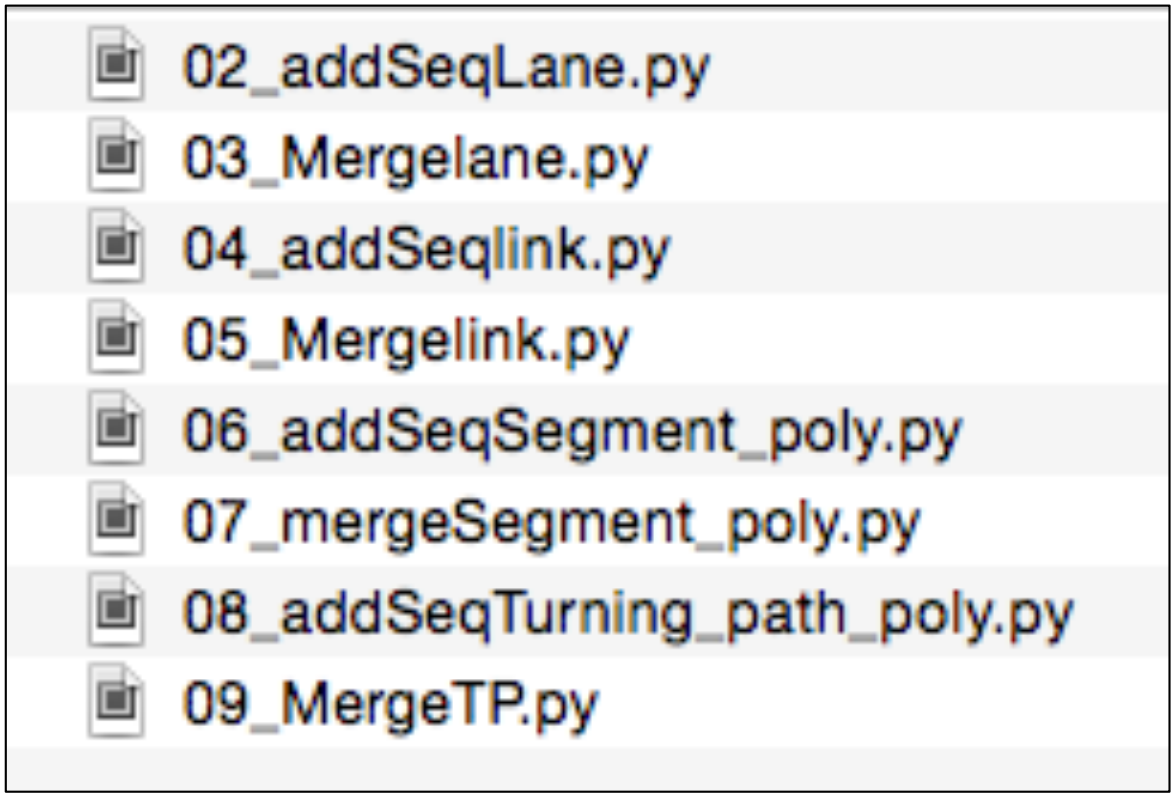
- Right click layers → save as → ESRI Shapefile → specify the location
- Save the files in
  - /home/smartfm/Documents/working_folder/edited_shp_files
- Check if the data is saved properly

Run Python Script to format and Merge

Exported CSV file will be added with sequence numbers and Z axist

1. sh csv_file_edit.sh

It will run the following scripts and generate xxx_polyline.csv files



- CSV files
- python files and
- .sh files

should be in the same folder

Import Data to Database

Run import_supply_data function with ‘schema_name’ and ‘path for the CSV file’

Go to PgAdmin → Public schema → function → import_supply_data(`schema_name`, `path`)


```
Select public.import_supply_data(‘supply’,
‘/home/smartfm/Documents/working_folder/edited_csv_files/’)
```

Validate supply data (SQL)

Open “SQL SCRIPTS” file and copy and paste the SQL commands in PgAdmin

Validate supply data (SQL)

Run SQL validating scripts in PgAdmin

- *Open PgAdmin*
- *Open SQL SCRIPTS file*
- *Copy and paste the sql script in PgAdmin*
- *Run each of the scripts one by one*

Validate supply data (Python)

Run Python validating scripts

Python2.7 01_Supply_Check_RoadNetwork_Consistency
Example of road network data consistency checking: lane has no downstream turning path, lane id ordering, segment's sequence, turning path connectivity, duplicate turning group, missing lane connector, etc..

Python2.7 02_Find_Closed_Nodes.py
First functionality, find the closed, loop, sink and source nodes, then produce nodes_new.csv.

Python2.7 03_Supply_Check_Facilities.py
Example of checks: invalid bus stop's segment, taxi stand' section offset is longer than segment's length, missing terminal_node data for the bus terminal and etc..

04_Supply_Check_Facility_Train.py.
Example of checks: invalid train platform, shared block among train platform, shared platform by multiple train lines and etc..

Link default travel time

SQL Scripts……..

```
with temp1 as(SELECT id, link_id, (length/(max_speed/3.6)) as travel_time
FROM supply.segment)
INSERT INTO supply.link_default_travel_time
SELECT link_id, 'Car' As travel_mode, '00:00:00' AS start_time,
'23:59:59' AS end_time, SUM (travel_time)
FROM temp1
GROUP BY link_id
ORDER BY link_id;
```

Visualizing the road-network

- cd /home/smartfm/simmobility-prod/dev/Basic

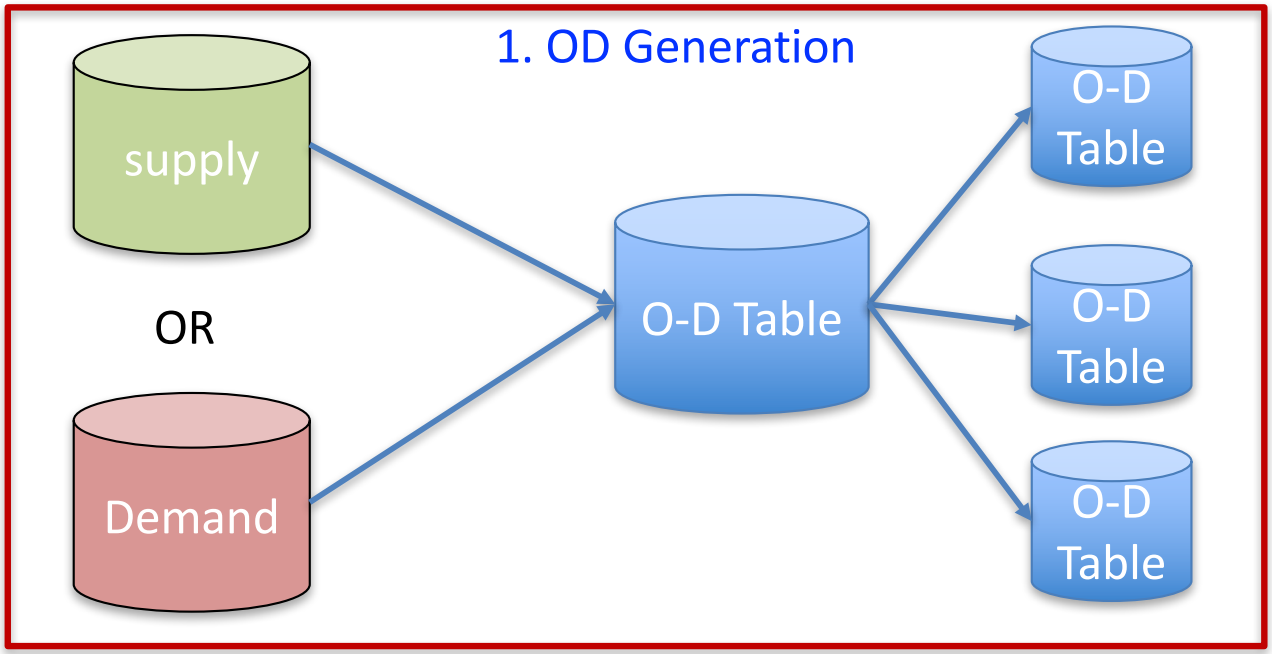
- Release_mid/SimMobility_MidTerm.xml data/simulation.xml data/simrun_MidTerm.xml

➔ It will create out.txt

- Run DynaVIS to load out.txt

Cars which are roaming may traverse through the new link

Pathset Generation



2. Setup Configuration file

- simulation.xml
- Simrun_MidTerm.xml
- Pathset_config.xml

3. Run Pathset Generation

4. Import the data in DB

5. Index the table in DB

Pathset Generation

1. O-D Generation

➢ If there is a change in the nodes, please re-generate node OD table in the database: routechoice.pvt_od

SQL query:

'''SELECT a.id as origin, b.id as destination FROM supply.node a LEFT JOIN supply.node b ON a.id != b.id'''

2. Setup configuration files:

➢ **simulation.xml**

- ❑ Reduce simulation runtime to 1 hour (line 94)

➢ **Simrun_MidTerm.xml**

- ❑ Run in "supply" mid_term_run_mode (line 8)
- ❑ Use Store get_person_between_zero_demand() (line 65)
- ❑ Update pathset_config_file value (line 133)
- ❑ Ensure PT controllers are off (line 175-194)

Pathset Generation

➢ pathset_config.xml

- ❑ Ensure pathset > private_pathset enabled, and mode = "generation" (line 3-5)
- ❑ Update to appropriate thread_pool size (line 4)
- ❑ Update od_source table / list of ODs to generate pathset on (line 6)
- ❑ Update bulk_generation_output file / output pathset file (line 7)
- ❑ Update other pathset parameters, if desired (line 12-19)

➢ 3. Run Pathset Generation:

- ❑ Use the same command to run SimMobility mid-term
- ❑ Ensure pointing to correct file paths
- ❑ Suggest to output console prints to a txt file for debugging if required

'''Release_mid/SimMobility_Medium data_pvtpathset/simulation.xml data_pvtpathset/simrun_MidTerm.xml > data_pvtpathset/xxx.txt'''

❑ Pathset generation successful when

"Private traffic pathset generation done"

Pathset Generation

➢ 4. Import Pathset in Database:

- ❑ Use the same command to run SimMobility mid-term
- ❑ Ensure pointing to correct file paths
- ❑ Suggest to output console prints to a txt file for debugging if required

'''\copy routechoice.pvt_path_netedit
from /home/smartfm/simmobility-prod/dev/Basic/pvt-pathset.csv with csv
➢ *data_pvtpathset/xxx.txt'''*

➢ 5. Import Pathset in Database:

CREATE INDEX idx_pvt_path_netedit
ON routechoice.pvt_path_netedit USING btree
(pathset_origin_node, pathset_dest_node);'''

Simulation with Edited Network