Cross-Resolution Modeling with NeXTA/DTALite

Examples with Portland and Tucson

Prepared by:
Jeffrey Taylor (jeffrey.taylor.d@gmail.com)
Last Revised: October 8, 2012

Please feel free to send any questions, feedback, and corrections to Jeffrey Taylor (jeffrey.taylor.d@gmail.com) or Dr. Xuesong Zhou (zhou@eng.utah.edu) by adding comments in this document and including the file as an attachment.

Copyright (C) 2012 Jeffrey Taylor.
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in www.gnu.org/licenses/fdl.html.

Table of Contents

Table of Figures

Example 1: Portland Network

This example provides a step-by-step process to guide the user through converting the Portland regional travel demand model network for use in NeXTA/DTALite, with additional instructions for performing subarea analyses with Synchro and VISSIM. The Portland regional travel demand model network used in this example was provided by Metro in Portland, Oregon. The model encompasses the Portland metropolitan area contained within Multnomah, Washington, and Clackamas counties in Oregon, as well as a portion of Clark County in Washington, with 43122 links, 14721 nodes, and 2162 zones. This network was originally assembled from NAVTEQ data for use with dynamic traffic assignment applications.

The conversion tools offer the ability to convert networks for modeling at different spatial and temporal resolutions. However, this type of conversion often results in problems with network representation and data inconsistencies. The examples in this section follow a specific methodology for maintaining network and input data integrity between network models to support analysis at mesoscopic and microscopic modeling resolutions. This methodology is detailed below.

Step 1: Export network from regional travel demand model
Step 2: Import network into NeXTA from regional travel demand model
Step 3: Read demand data into NeXTA from regional travel demand model
Step 4: Run assignment with DTALite to equilibrium
Step 5: Cut a subarea within the larger model for more detailed analysis
Step 6: Prepare field data for ODME (Origin-Destination Matrix Estimation)
Step 7: Run ODME using field data for calibration
Step 8: Export to Synchro™/VISSIM™ for signal optimization/microscopic analysis

The process begins with simply importing the regional travel demand model network into NeXTA using the network conversion tool, and importing/reading previously-prepared demand tables. The remaining steps demonstrate how to cut a subarea in the network for analyzing smaller regions, how to estimate new demand tables/matrices to better represent observed conditions in the network, and how to export the subarea network to Synchro™ and VISSIM™.

Step 1: Export network from regional travel demand model

Since the Portland regional travel demand model was coded in VISUM, the first step in the network conversion process is to create a set of shapefiles describing the roadway network to be imported into NeXTA. This is accomplished using export functions in VISUM, and can be divided into two internal steps:

1. Load the network in VISUM, and export the network as shapefiles
2. Export demand tables/matrices

1) Load the network in VISUM, and export the network as shapefiles

In this example, the Portland regional travel demand network was coded in VISUM and must be exported as a set of shapefiles. First, the network is loaded in VISUM.

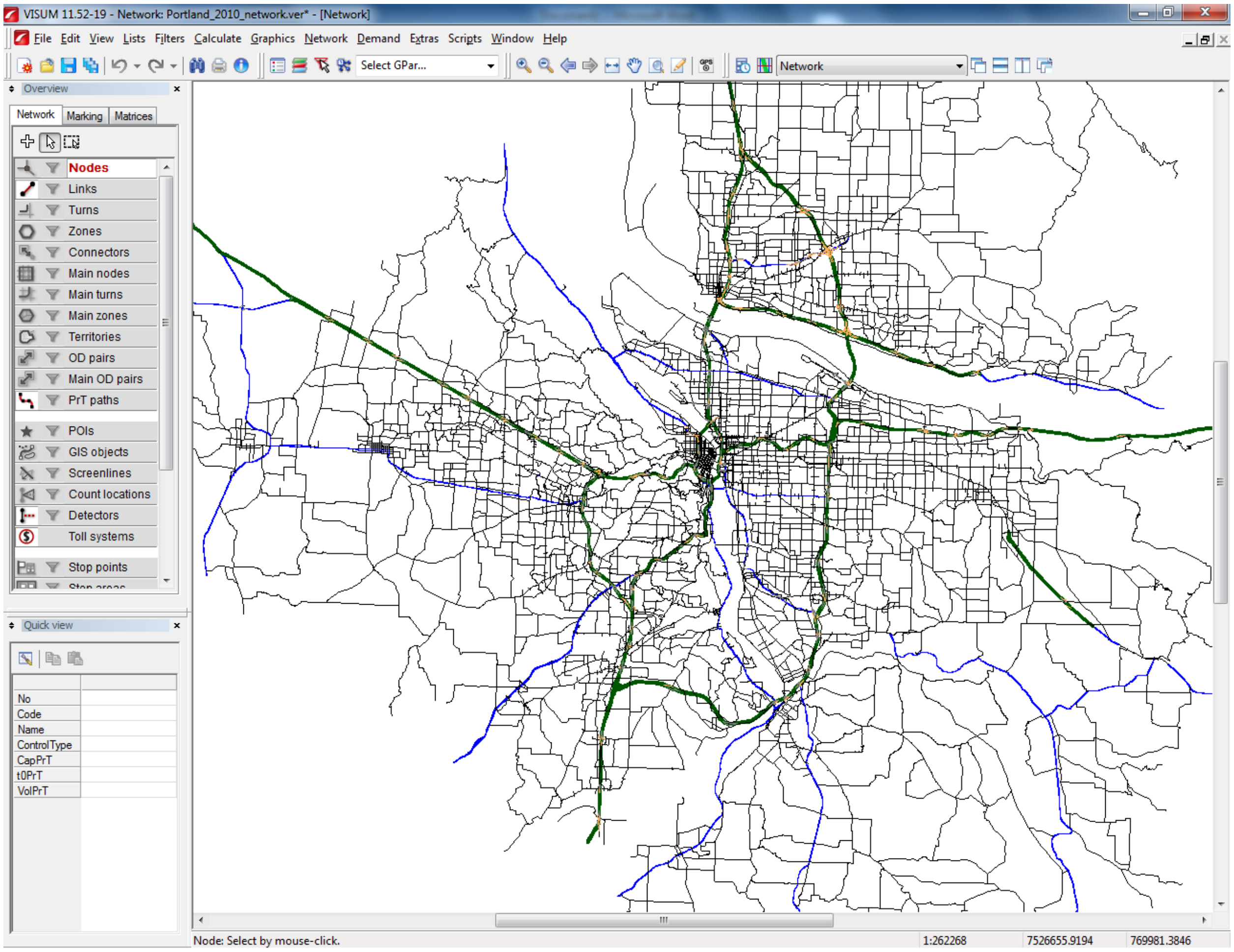

Figure 1: Portland regional travel demand model loaded in VISUM

By using VISUM's export function **(File > Export > Shapefile)**, the network is split into its components, and saved as multiple separate shapefiles. Since the user can choose which layers to export, the user should select the node, link, zone, zone centroid, and connector layers to ensure that the conversion process can successfully create a new network. While it is recommended that the user change the export options to save the links as directed links to minimize any potential problems in the conversion process, the appropriate setting will depend on the network coding format. NeXTA can import networks with both directed and non-directed links.

The user can also identify which columns will be exported with the shapefiles, which may be helpful to minimize confusion when working with shapefiles with large database files, or when working with multiple scenarios in the same network files.

Note: VISUM will export the shapefiles in the network's current coordinate system. To simplify the conversion process, the user may change the network coordinate system before performing the export process described below. Please see Intermediate Step: Change map projection to WGS84 for more information. Exporting to shapefiles in VISUM requires the GIS Interface Shape add-on module. Please refer to the VISUM user manual to identify how to enable this functionality if it is not available.

Once finished selecting export options, select “OK” at the bottom of the dialog box.

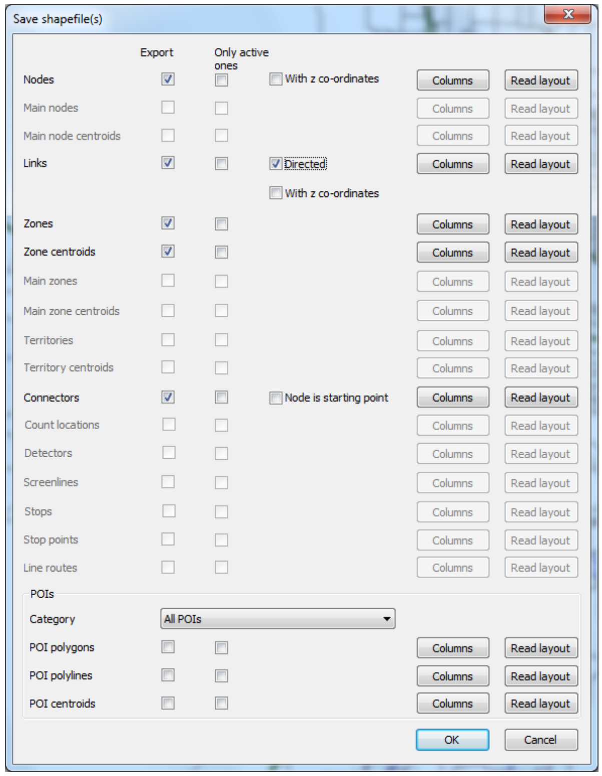

Figure 2: Shapefile export options in VISUM

2) Export demand tables/matrices

Click on the Matrices tab in the Overview toolbar to display the demand matrices. The zone matrices are compatible with the demand definition in NeXTA, so they are the main focus of this effort. Right click on the matrix the user desires to export, and select “Save to File…”, as shown in the figure below.

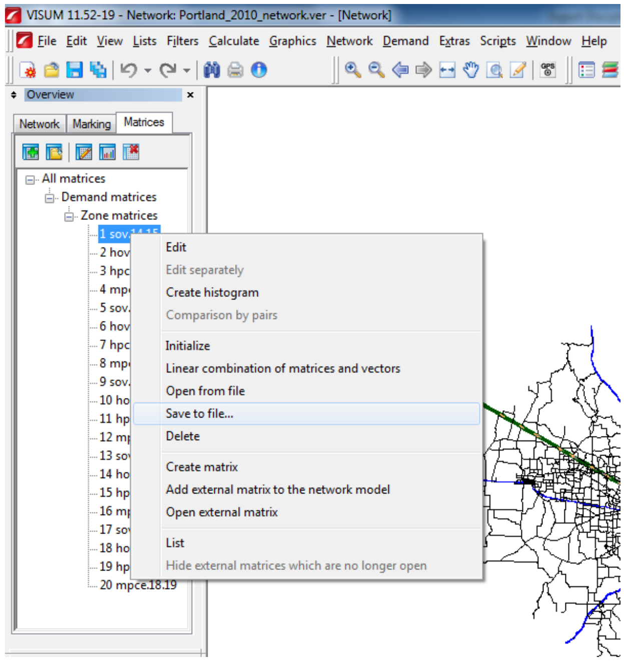

Figure 3: Saving demand matrices in VISUM to be imported in NeXTA

Choose a location to save the matrix file at the “Save: Matrix” context menu. Afterward, a new dialog box opens with export options, as shown below. VISUM offers multiple formats for exporting demand matrices, but it is recommended that the user use the “Format O” option when preparing tables for export to NeXTA. Format O is a common 3-column format that can be easily read by NeXTA. NeXTA also supports the standard VISUM “Format V” demand matrix format, which takes the form of a table with size Origin x Destination. The user should reference their VISUM User Manual for more information regarding different demand matrix formatting types.

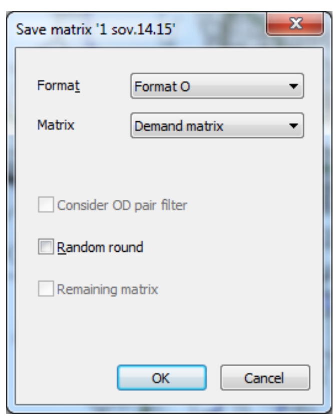

Figure 4: Matrix export options in VISUM

After selecting the export options, select “OK” at the bottom of the dialog box.

Exporting the demand matrices produces 20 demand tables describing the number of trips between zones for different demand types (SOV, HOV, Heavy Truck, and Medium Truck) for the time period between 2pm and 7pm.

Intermediate Step: Change map projection to WGS84

The original VISUM network used the NAD83 coordinate system, and thus required conversion to the WGS84 system before being imported into NeXTA. This process was completed using projection tools in ESRI's ArcGIS software, although this could also have been accomplished by changing the map projection in VISUM before export (explained below). Please note that reverse direction fields in the zonal connector shapefile may cause problems when using ArcGIS to change the map projection, since the field precision may not be supported (issue in VISUM).

Changing the map projection to WGS84 can be accomplished using two separate methods – one before export, and one after export from the original source software.

Option 1: Change the network coordinate system in VISUM before export

Option 2: Change the map projection for each exported shapefile in a GIS software package

Option 1: Change the network coordinate system in VISUM before export

When creating a network in VISUM, the user defines a network coordinate system by changing the network parameters. VISUM, by default, may not use a pre-defined coordinate system unless those parameters are changed. For this reason, it is important to make sure that the shapefiles are exported in a pre-defined coordinate system, since it may be very difficult to convert those shapefiles to another pre-defined coordinate system (i.e., WGS84) when they have been created using VISUM's own default coordinate system. Using the same technique, the user can simplify the conversion process by changing the network coordinate system to WGS84 in VISUM before exporting the shapefiles. Please refer to the VISUM User Manual for instructions about how to change the network coordinate system in VISUM.

Option 2: Change the map projection for each exported shapefile in a GIS software package

If the network has already been exported from the originating software product (i.e., VISUM), and the map projection is not defined as WGS84, the user may use a third-party GIS software product to perform the map projection conversion. A number of different GIS products are available with map projection conversion tools, with several open-source options available: Quantum GIS, MapWindow GIS, uDig, etc. Please refer to the manual of the user's preferred GIS software for instructions to change the map projection for the exported shapefiles.

Step 2: Import network into NeXTA from regional travel demand model

The second step in the network conversion process is to use NeXTA's network import tool to convert the network shapefiles, created in the previous step, into a network which can be used with NeXTA and DTALite. This process reads the spatial data stored in each specified shapefile, along with the corresponding data stored in the database (DBF) file, to create the input CSV files used by NeXTA and

DTALite. In order to interpret the database (DBF) files for conversion, a configuration (INI) file is used to map field names between the shapefiles and the standard data format used by NeXTA.

The network import process is divided into three internal steps:

1. Prepare INI configuration file & attribute files for conversion
2. Use NeXTA's import network tool to convert the network
3. Save the new network as a new project file

1) Prepare INI configuration file & attribute files for conversion

The first step in converting the network is to create a configuration file (*.INI) in the folder containing the exported shapefiles. NeXTA uses this user-defined configuration file (*.INI) to associate/map fields in shapefile DBF files to NeXTA/DTALite's data format, allowing NeXTA to read the network geometry from shapefiles and create a NeXTA/DTALite-compatible transportation network project file.

The configuration file is divided into different sections depending on the type of data to be imported. The first section describes general model attributes and import options. The remaining sections are used to describe the different types of network objects which can be imported. Separate sections are used to import links and nodes, with optional sections for importing zones, zone centroids, and zonal connectors. Detailed descriptions are provided for each critical section below. Also, please refer to the import_configuration.ini file, found in the Portland_Example folder, for the detailed configuration file used for conversion.

Model attributes configuration settings (with reasoning/explanation):

Variable Assignment	**Description/Explanation**
units = MI	The original network describes distances in units of miles.
direction_field =0	"0" indicates that NeXTA does not use a "direction" field to interpret the link shapefile. (No direction field, use 0)
control_type_field=1	"1" indicates that NeXTA will read control type information from the node shapefile.
reserve_direction_field=0	The links were exported as directed links, meaning that there is no reverse direction data in the shapefile. "0" indicates that NeXTA will read direction-specific information.
offset_link=1	"1" indicates that NeXTA will add space between directional links to aid visualization.
link_capacity_flag=1	"1" indicates that the link capacity is described on a per-link basis, which is the default in VISUM. Lane capacity is calculated by dividing by the number of lanes.
number_of_lanes_for_two_way_links=0	"0" indicates that lanes on two-way links are described using the number of lanes in each direction.
use_optional_centroid_layer=1	"1" indicates that there is an additional shapefile to read which only contains zone centroids.
use_optional_connector_layer=1	"1" indicates that there is an additional shapefile to read which only contains zonal connector links.

Node import configuration settings (with reasoning/explanation):

Variable Assignment	Description/Explanation
reference_file_name = New_Portland_node_WGS84.shp	Identifies “New_Portland_node_WGS84.shp” as the shapefile which describes the nodes in the network.
node_id = NO	Identifies “NO” as the name of the field which contains unique node numbers.
control_type = CONTROLT_1	Identifies “CONTROLT_1” as the name of the field which contains control type id numbers mapped to the associated node numbers.

Link import configuration settings (with reasoning/explanation):

Variable Assignment	Description/Explanation
reference_file_name = New_Portland_link_WGS84.shp	Identifies “New_Portland_link_WGS84.shp” as the shapefile which describes the links in the network.
from_node_id = FROMNODENO	Identifies “FROMNODENO” as the name of the field which describes the starting node number for one direction on a link.
to_node_id = TONODENO	Identifies “TONODENO” as the name of the field which describes the ending node number for one direction on a link.
link_id = NO	Identifies “NO” as the field name containing unique link numbers.
length_in_mile = 'LENGTH'	Identifies “LENGTH” as the name of the field describing the length (in miles) of each link.
link_type = TYPENO	Identifies “TYPENO” as the name of the field which describes the functional classification (equivalent to link type in NeXTA/ DTALite) for the directional link.
number_of_lanes = NUMLANES	Identifies “NUMLANES” as the name of the field which describes the number of lanes on the directional link.
lane_capacity_in_vhc_per_hour = CAPPRT	Identifies “CAPPRT” as the name of the field which describes the link capacity on the directional link. This field is converted to lane capacity by the “link_capacity_flag” variable.
speed_limit_in_mph = V0PRT	Identifies “V0PRT” as the name of the field which describes the free-flow speed on the directional link.

Zone import configuration settings (with reasoning/explanation):

Variable Assignment	Description/Explanation
reference_file_name = New_Portland_zone_WGS84.shp	Identifies “New_Portland_zone_WGS84.shp” as the shapefile which describes the zones in the network.
zone_id = NO	Identifies “NO” as the name of the field which contains unique zone numbers.

Zone Centroid import configuration settings (with reasoning/explanation):

Variable Assignment	**Description/Explanation**
reference_file_name = New_Portland_zone_centroid_WGS84.shp	Identifies “New_Portland_zone_centroid_WGS84.shp” as the shapefile which describes the zone centroids in the network.
node_id = NO	Identifies “NO” as the name of the field which contains unique node numbers.
TAZ = NO	Identifies “NO” as the name of the field which contains unique zone numbers mapped to the node number. Zone centroids are mapped to nodes with one-to-one mapping in this network, so the variables have the same name.

Zonal Connector configuration settings (with reasoning/explanation):

Variable Assignment	**Description/Explanation**
reference_file_name = New_Portland_connector_WGS84.shp	Identifies “New_Portland_connector_WGS84.shp” as the shapefile which describes the zonal connector links in the network.
zone_end = ZONENO	Identifies “ZONENO” as the name of the field which describes the zone number attached to one end of the zonal connector.
node_end = NODENO	Identifies “NODENO” as the name of the field which describes the node number attached to one end of the zonal connector.
length_in_mile = 'LENGTH'	Identifies “LENGTH” as the name of the field describing the length (in miles) of each zonal connector.
default_number_of_lanes =2	Identifies a default number of lanes (2) to be used when importing the zonal connector.
lane_capacity =10000	Identifies a default lane capacity to be used when importing the zonal connector.
default_speed_limit =60	Identifies a default free-flow speed to be used when importing the zonal connector.
default_link_type_for_connector =99	Identifies a unique link type number to be assigned to the link when importing the zonal connector.
default_direction =0	Identifies the default direction of the zonal connector. “0” and “2” indicate two-way links, “1” and “-1” indicate specific directions.

<u>2) Use NeXTA’s import network tool to convert the network</u>

The next step is to import the network data into NeXTA. Starting with a new, empty network project in NeXTA, the conversion process is initiated by selecting the appropriate tool in the Import Menu by going to **File > Import > Transportation Network Data Set**.

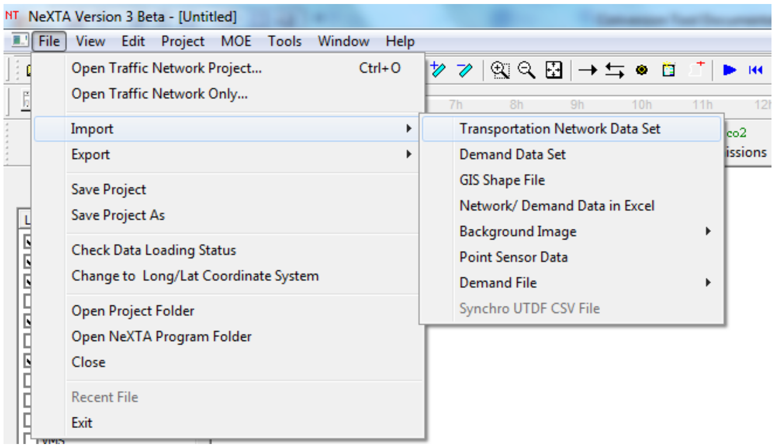

Figure 5: Network Import Menu location in NeXTA

Select the appropriate INI configuration file created in the previous step, and click “Open”.

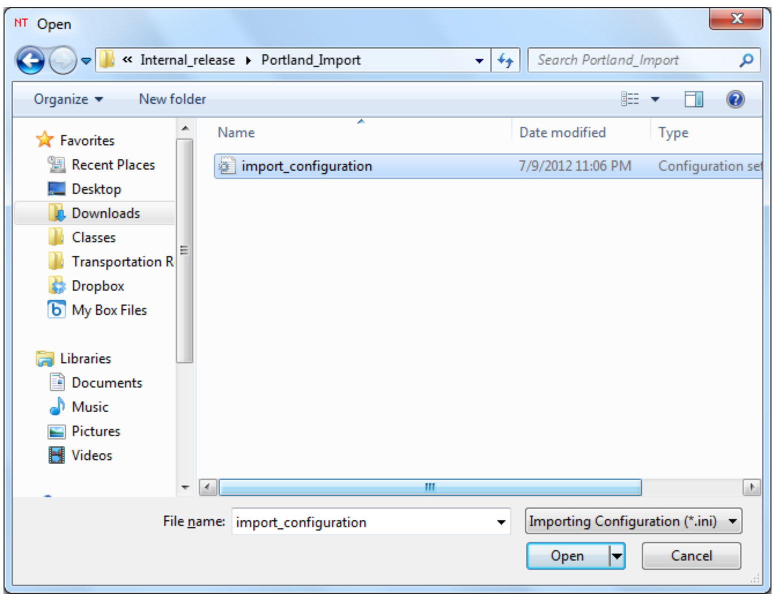

Figure 6: Opening the import configuration INI file in NeXTA

After the conversion process is complete, a File Loading Status window opens to display the results of the conversion process. Make sure to note the number of links, nodes, and zones (if any were selected for import) against the number of records in the shapefiles to make sure the process was completed successfully (no missing records).

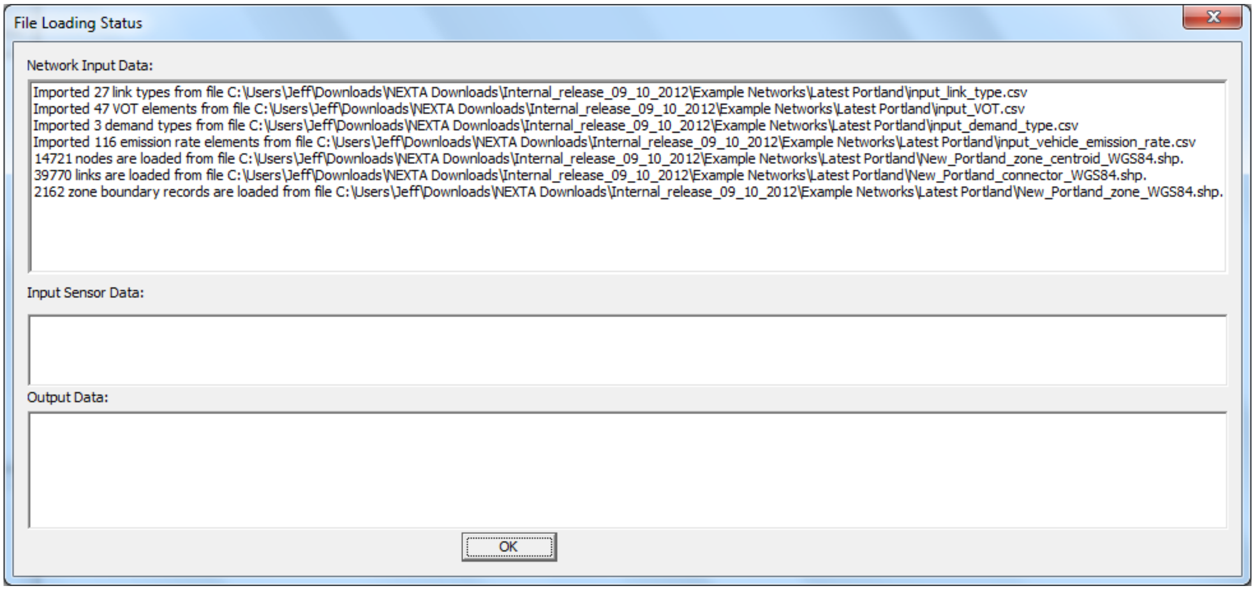

Figure 7: File loading status window displays import results after completion

The imported network should look similar to Figure 8 below.

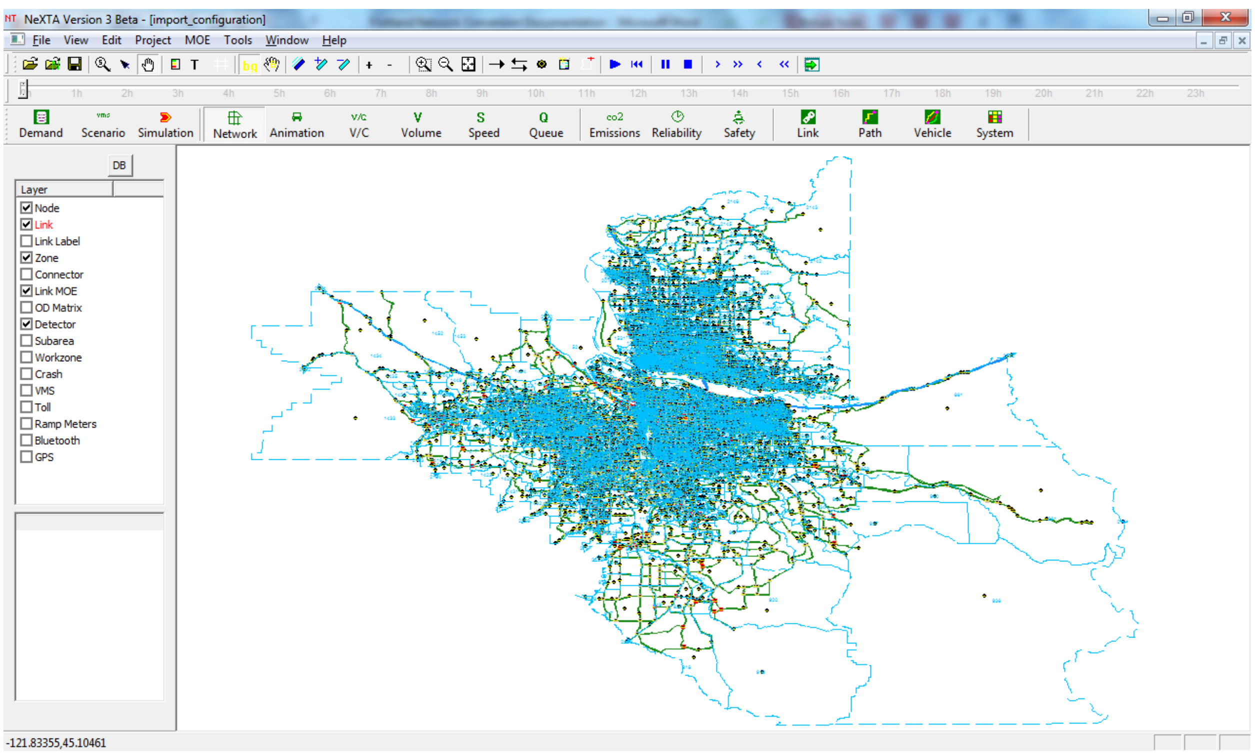

Figure 8: Portland regional network imported into NeXTA/DTALite

<u>3) Save the new network as a new project file</u>

Finally, save the network as a new Transportation Network Project (*.tnp) file to write the input tables in the project folder.

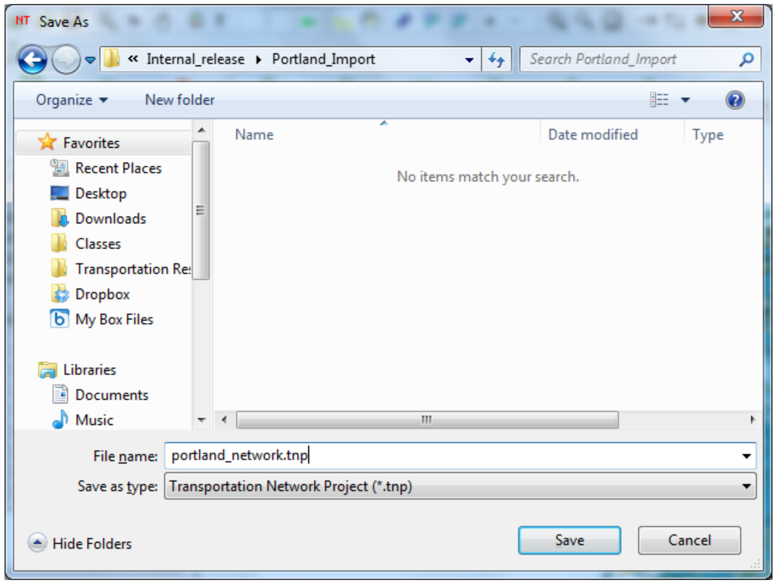

Figure 9: Save the new network to create a new NeXTA/DTALite project file

Step 3: Read demand data into NeXTA from regional travel demand model

The new demand data handling in NeXTA/DTALite has significantly reduced the complexity involved in converting demand data between different software applications. In the current implementation, individual demand tables are read by NeXTA/DTALite when they are specified in the input_demand_meta_data.csv file. Thus, the user must only edit the demand meta-data table to read the MTX files exported from VISUM for the Portland regional travel demand model.

This can be accomplished using a simple 5 step procedure for each demand data file to be used in the simulation, as described below with examples in parentheses for the Tucson regional network.

1. Specify the demand file name (e.g., sov_14_15.mtx) and format (e.g., column)
2. Specify the number of lines in the demand file to be skipped by DTALite (8 for MTX file)
3. Indicate whether subtotals are present in the last column (zero for none)
4. Specify the loading start time and end time for the demand file (840 to 900, or 2PM to 3PM)
5. Specify the demand types associated with the demand file (only demand_type1)

Step 4: Run assignment with DTALite to equilibrium

Since the objective of this example is to perform more detailed analyses, a subarea will be used to create a smaller network which can be used for microsimulation analysis. Accurately representing the origin-destination flow in a subarea requires creating external zones at the subarea boundary. When the subarea is created, the path flows are aggregated at the links which cross the subarea boundaries, assigning the aggregated trips to the external zone created at the end of those links. However, no path flow data exists without first running the full network simulation. As a result, the preliminary step performed before creating a subarea network is to create the network path flows by running the DTALite assignment engine to equilibrium.

Now that the network and demand are imported into a format which is compatible with NeXTA/DTALite, running the DTALite assignment engine only requires editing the simulation settings in the input_scenario_settings.csv file, and initiating the assignment engine. To run the assignment engine in NeXTA/DTALite, select Simulation from the Toolbar at the top of the screen.

For this example, the simulation is run for 15 iterations with 1,115,689 agents, requiring 1 hour 13 minutes computational time on an Intel Core I7 2760QM (2.4-3.5 GHz quad core) with 32GB RAM. Assignment resulted in an average travel time of 26.18 minutes with an average trip length of 7.72 miles.

Step 5: Cut a subarea within the larger model for more detailed analysis

Performing more detailed analyses with large networks like the Portland regional network often requires creating a subarea network. A subarea network is a self-contained sub-network defined by a subarea boundary, where objects outside the subarea boundary are cut from the network, and new external zones are created at the subarea boundary where links cross the boundary. These external zones are necessary to preserve the origin-destination traffic flow through the subarea.

The subarea creation process is divided into four internal steps:

1. Create a subarea boundary in NeXTA
2. Use NeXTA's Subarea Cut tool to clip the network
3. Convert zonal connectors to side-streets within the subarea
4. Save the new subarea network as a new project file

1) Create a subarea boundary in NeXTA

In order to cut a subarea, a subarea boundary must first be defined. Using the **Create Subarea** tool in NeXTA, a subarea boundary was drawn around the 185th Avenue evaluation area used for this test case. After drawing the boundary, the links and nodes within the boundary are highlighted by NeXTA, as shown in the figure below.

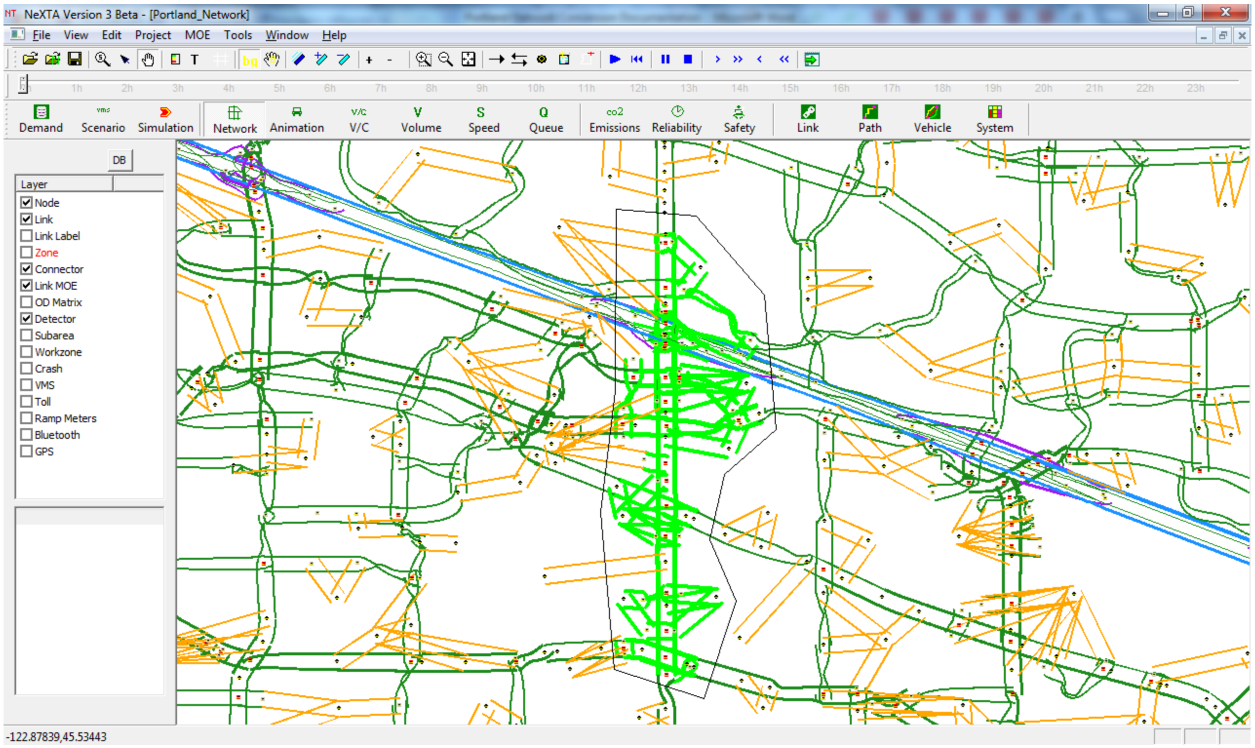

Figure 10: Subarea boundary selection for Portland 185th subarea

2) Use NeXTA's Subarea Cut tool to clip the network

Next, use the **Subarea Cut** tool in NeXTA to delete all of the network objects (nodes and links) outside of the subarea boundary by going to **Tools -> Subarea Cut**. NeXTA will display a dialog showing the number of links, nodes, zones, OD pairs, and subarea path records are extracted for the subarea analysis. The resulting subarea network should look similar to the figure below.

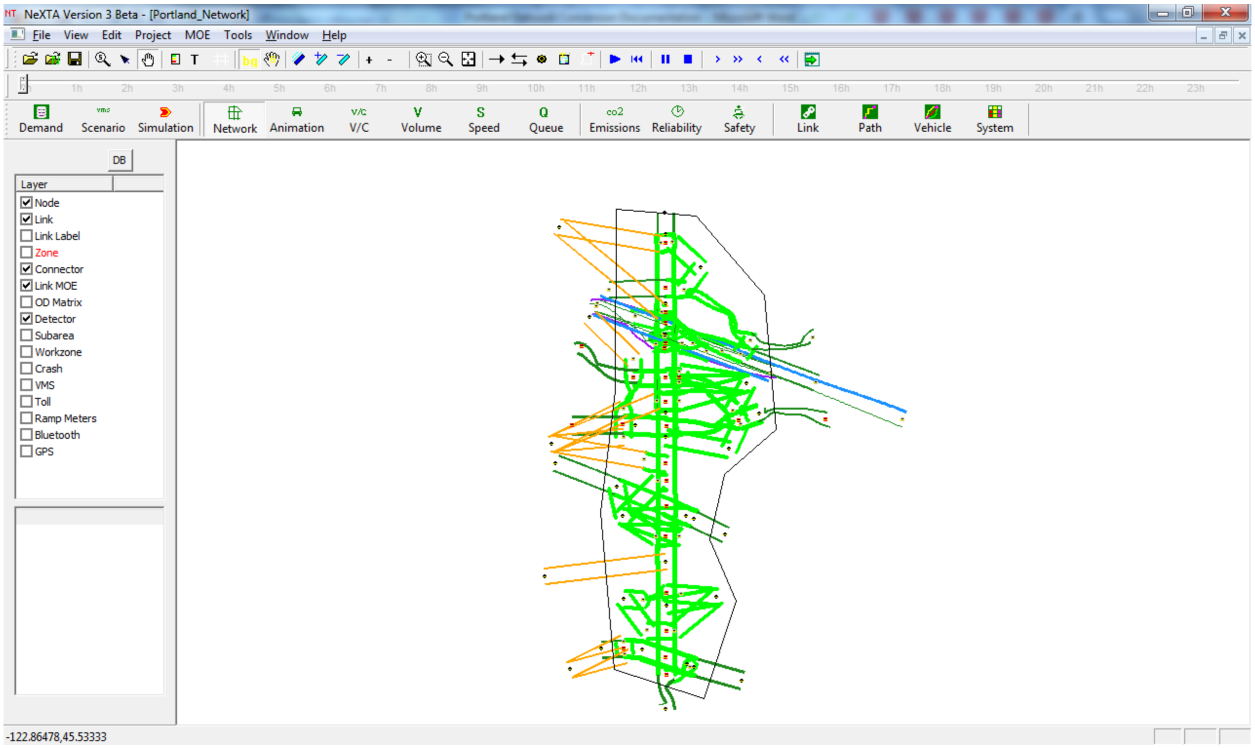

Figure 11: Clipped subarea for the Portland 185th subarea

3) Convert zonal connectors to side-streets within the subarea

Lastly, use the **Generate Physical Zone Centroids on Road Network** tool to convert the zonal connectors to side streets within the network by going to **Tools > Generate Physical Zone Centroids on Road Network**. This tool replaces zone centroids with additional nodes so that no paths can be routed through a zone centroid. While DTALite cannot use paths through zone centroids, Synchro and VISSIM can use those paths, making using this tool a very important step in the process toward creating a new Synchro and/or VISSIM network. The resulting network created by using this tool should look similar to the network shown in the figure below.

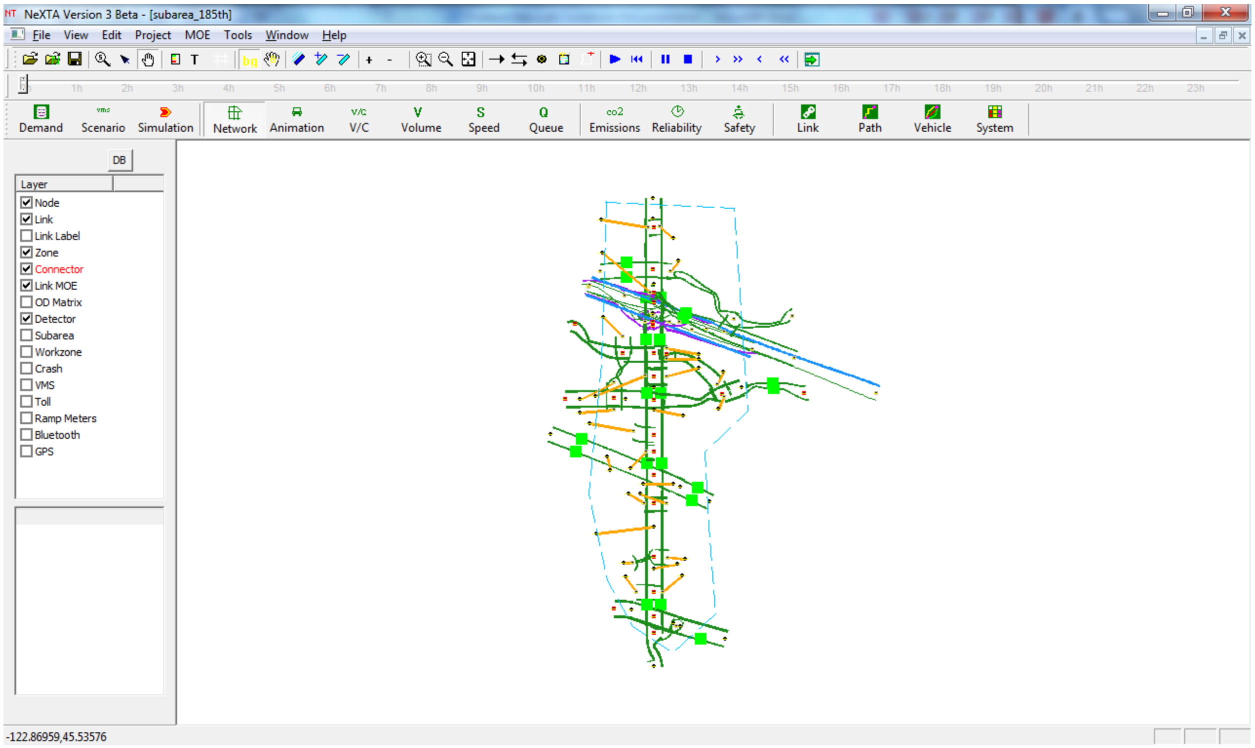

Figure 12: Results from converting zonal connectors to side-streets in the Portland 185th subarea

4) Save the new subarea network as a new project file

The last step is to save the new subarea network in a new project file. Go to **File > Save Project As**, and select a new location for the new subarea project TNP file.

Step 6: Prepare field data for ODME (Origin-Destination Matrix Estimation)

Origin-Destination Matrix Estimation is a technique used to adjust demand patterns in a network to better approximate observed traffic conditions (e.g., time-dependent link volume). This is normally accomplished in an iterative process by assigning trips to paths in a network, comparing observed and simulated link volumes/counts, adjusting the input demand data, and moving to the next iteration where the trips are re-assigned. However, before running ODME, the field data must be prepared in a format which enables the ODME model to compare observed and simulated link volumes/counts.

DTALite's ODME model reads field data from the input_sensor.csv file, which the user must prepare before executing the ODME process. This input file uses a flexible format for reading multiple types of observed data in the network, including link volume, occupancy, speed, and travel time field data for specific locations and time periods, allowing for time-dependent ODME applications. The Portland 18th subarea field data was prepared from link volume counts collected at 21 locations on freeways and arterials in the subarea model, with hourly and 15-minute link volume counts. Their locations are represented as green squares in the figure below.

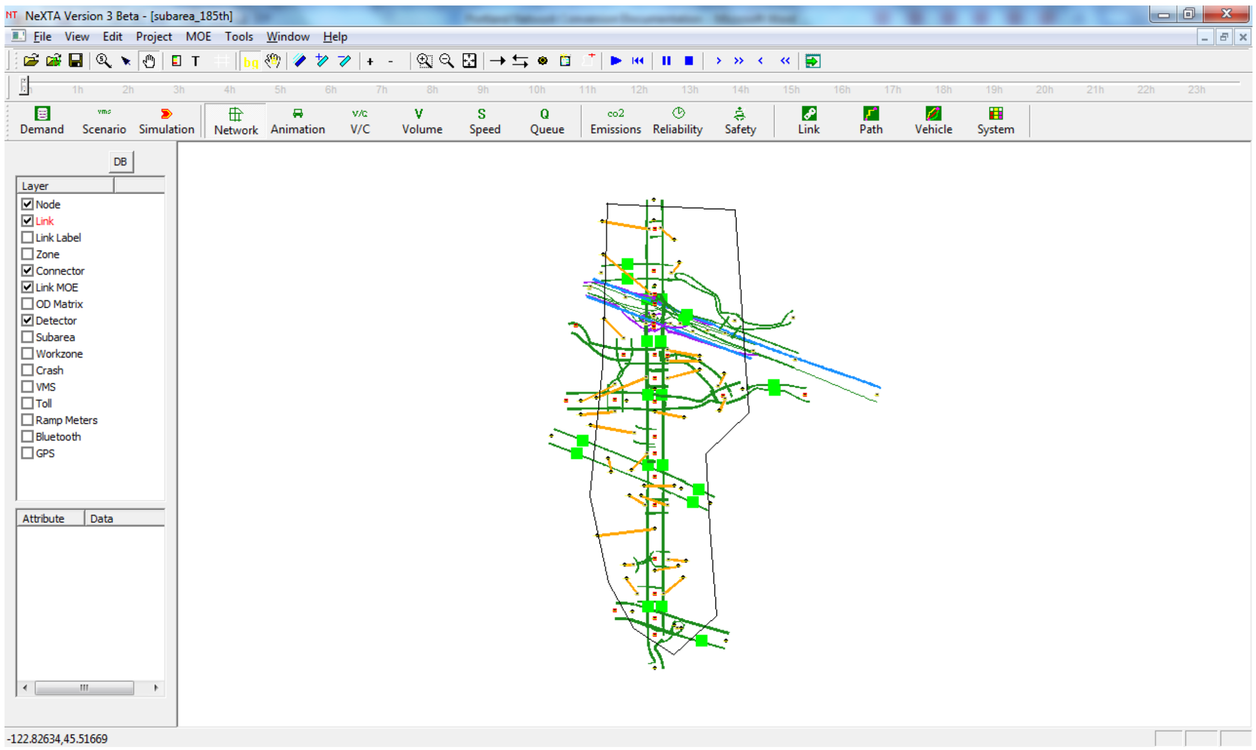

Figure 13: Subarea field data sensor locations for ODME in DTALite

Step 7: Run ODME using field data for calibration

This step-by-step example describes a process for delivering the network geometry and demand data, represented by OD path flows, to be used in microsimulation analysis. Since microscopic simulation relies upon detailed, accurate input demand data to best represent observed traffic conditions in the analysis subarea, it is important to make sure that the OD path flows closely align with these observed traffic conditions. This is accomplished using the ODME component in DTALite, which is used similarly to a network calibration tool in this procedure.

To enable ODME mode in DTALite, the user simply changes a flag variable in the input_scenario_settings.csv file, and adjusts the model parameters in the ODME_Settings.txt file. For this example, the assignment engine is specified to run for 5 iterations to arrive at a relatively stable operating condition, and then uses the ODME module to adjust path flows over the following 75 iterations. The ODME module in DTALite was set to adjust 5% of the OD demand at each iteration, allowing the model to run to completion faster without sacrificing solution quality. Time-dependent link volume/count data was used for calibration, allowing for more detailed OD flow refinements.

The two plots shown below compare the observed and simulated link volumes/counts at the subarea sensor locations. The initial equilibrium assignment (before ODME) produced link volumes that were relatively similar to the observed link volumes with R^2 = 0.55, although under- and over-estimation was observed at multiple locations. After running ODME for 80 iterations, the under- and over-estimation was reduced, and the R^2 improved to 0.79 over all observations.

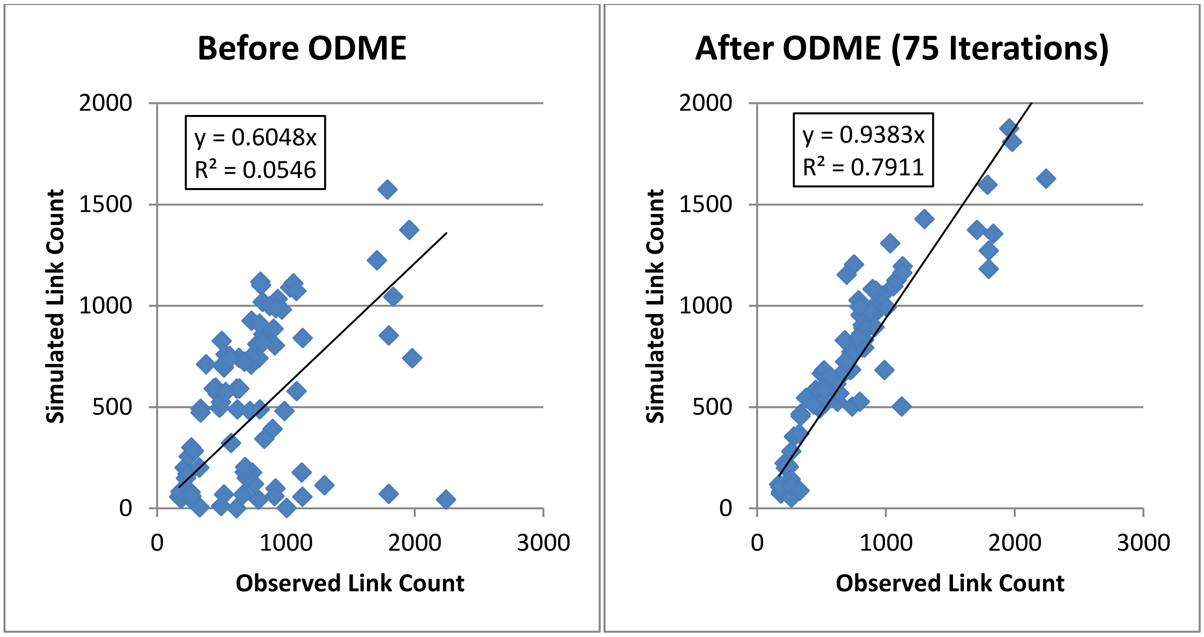

Figure 14: Origin-Destination Matrix Estimation (ODME) results for Portland 185th subarea

Step 8: Export to Synchro/VISSIM for signal optimization/microscopic analysis

After adjusting the demand data to better represent observed conditions in the subarea network, the subarea network is ready to be exported to Synchro™ and VISSIM™ for microscopic analysis. However, since NeXTA does not currently support importing signal timing data from the regional demand model (if any exists), initial signal phasing and timing is estimated using a Quick Estimation Method (QEM) approach, which is implemented as an automated, external spreadsheet. The procedure for exporting the subarea network for microscopic analysis is divided into three steps:

1. Use QEM to estimate initial signal phasing and timing
2. Export to Synchro using UTDF CSV format
3. Export to VISSIM using ANM format

1) Use QEM to estimate initial signal phasing and timing

An automated QEM spreadsheet is used to generate initial signal phasing and timings for the subarea network, which are exported to Synchro for analysis in the next step. To initiate the QEM spreadsheet calculations, go to **File > Export > Microscopic Network and Traffic Control Data**, and select Perform Quick Estimation Method (QEM) for Signals. NeXTA writes the geometry and volume information to the spreadsheet, the spreadsheet calculates appropriate phasing and timing data, and NeXTA reads that phasing and timing data back into NeXTA.

2) Export to Synchro using UTDF CSV format

Next, the network is exported into UTDF CSV files, which Synchro uses to create the subarea network through its internal import function. The imported network in Synchro is shown in the figure below.

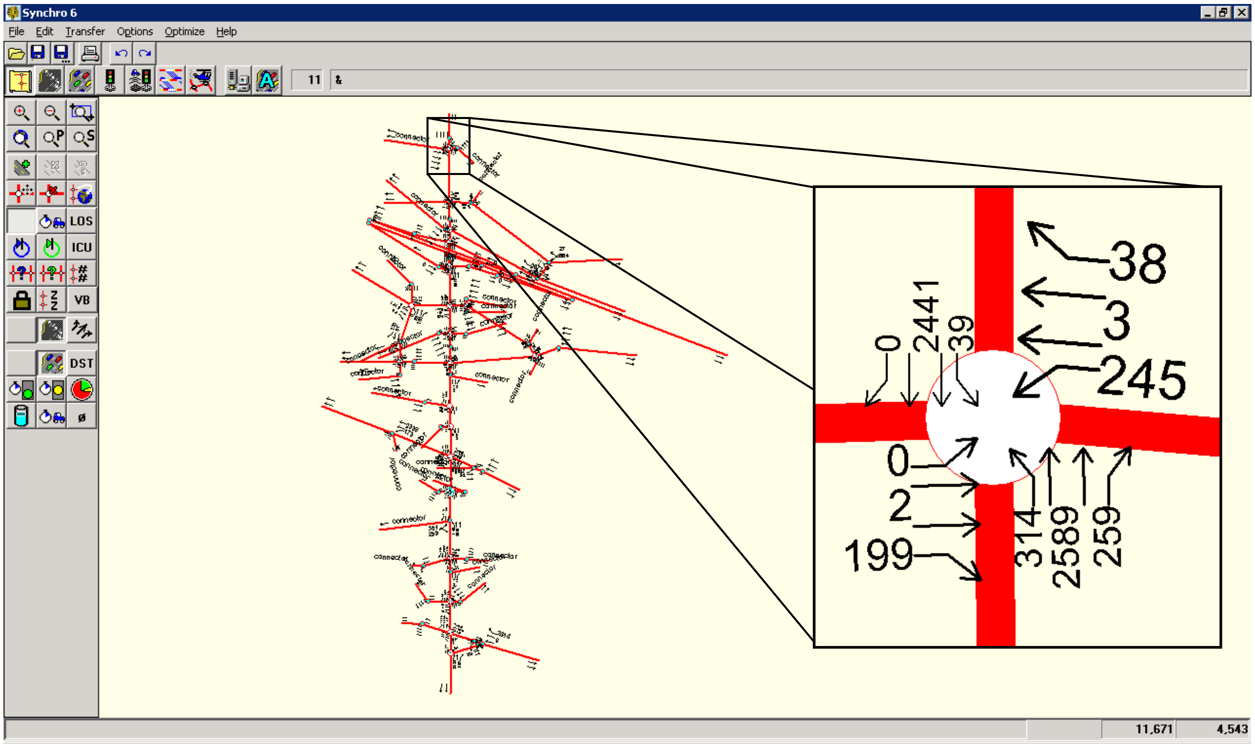

Figure 15: Portland 185th subarea network imported in Synchro from NeXTA

3) Export to VISSIM using ANM format

Lastly, the network is exported into the ANM VISSIM format, which VISSIM uses to create the subarea network through its internal import function. The imported network in VISSIM is shown in the figure below.

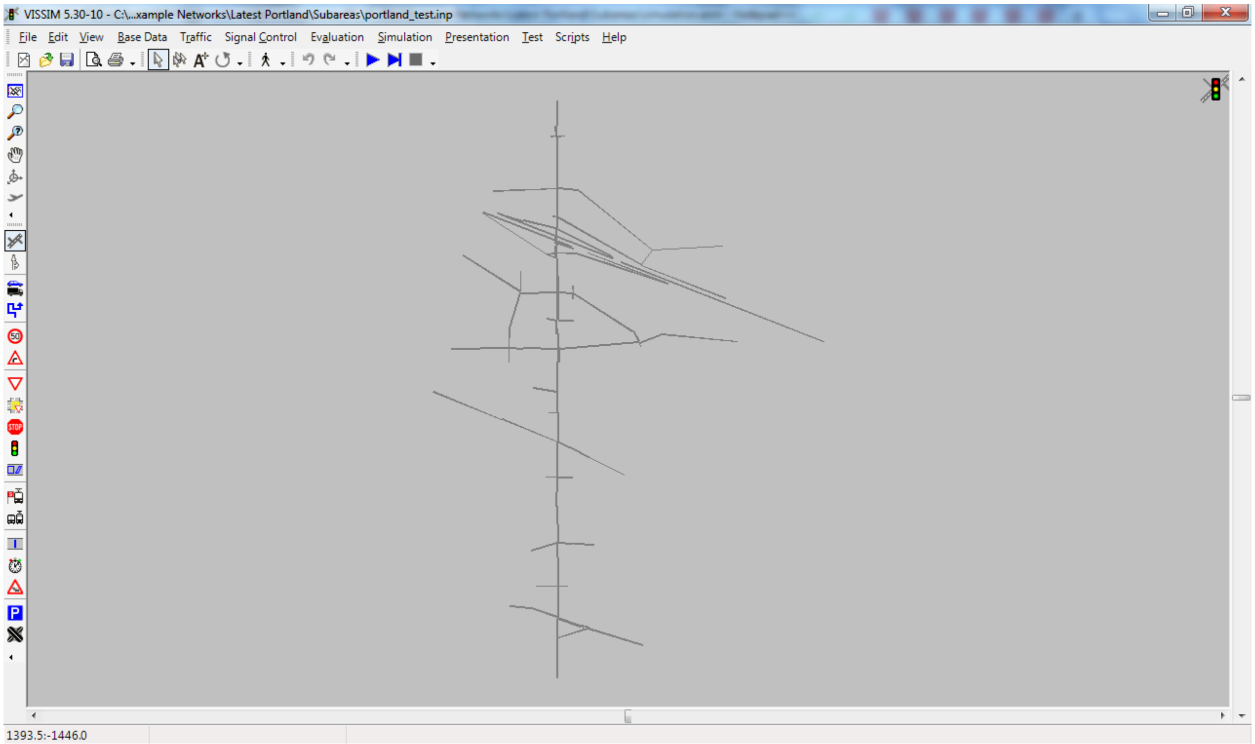

Figure 16: Portland 185th subarea network imported in VISISM from NeXTA

Example 2: Tucson Network

This example provides a step-by-step process to guide the user through converting the Tucson regional travel demand model network for use in NeXTA/DTALite, with additional instructions for performing subarea analyses with Synchro and VISSIM. The Tucson regional travel demand model network used in this example was provided by Kittelson & Associates, Inc. The model encompasses the majority of the Tucson metropolitan area contained within Pima County, Arizona with 12230 links, 4534 nodes, and 1121 zones.

The conversion tools offer the ability to convert networks for modeling at different spatial and temporal resolutions. However, this type of conversion often results in problems with network representation and data inconsistencies. The examples in this section follow a specific methodology for maintaining network and input data integrity between network models to support analysis at mesoscopic and microscopic modeling resolutions. This methodology is detailed below.

Step 1: Export network from regional travel demand model
Step 2: Import network into NeXTA from regional travel demand model
Step 3: Read demand data into NeXTA from regional travel demand model
Step 4: Run assignment with DTALite to equilibrium
Step 5: Cut a subarea within the larger model for more detailed analysis
Step 6: Prepare field data for ODME (Origin-Destination Matrix Estimation)
Step 7: Run ODME using field data for calibration
Step 8: Export to Synchro™/VISSIM™ for signal optimization/microscopic analysis

The process begins with simply importing the regional travel demand model network into NeXTA using the network conversion tool, and importing/reading previously-prepared demand tables. The remaining steps demonstrate how to cut a subarea in the network for analyzing smaller regions, how to estimate new demand tables/matrices to better represent observed conditions in the network, and how to export the subarea network to Synchro™ and VISSIM™.

Step 1: Export network from regional travel demand model

Since the Tucson regional travel demand model is coded in TransCAD, the first step in the network conversion process is to create a set of shapefiles describing the roadway network to be imported into NeXTA. This is accomplished using the shapefile export function in TransCAD. The resulting shapefiles describe the node and link layers in the network. After exporting the shapefiles, the demand matrices are exported to CSV files using the matrix export function in TransCAD. This produces 10 demand matrices describing the number of trips between zones for different demand types (drive alone, shared ride, bike, walk) and pricing types (general purpose, HOV, and pay/HOT).

Intermediate Step: Change map projection to WGS84

The original TransCAD network was used the NAD83 coordinate system, and thus required conversion to the WGS84 system before being imported into NeXTA. This process was completed using projection tools in ESRI's ArcGIS software.

Step 2: Import network into NeXTA from regional travel demand model

The second step in the network conversion process is to use NeXTA's network import tool to convert the network shapefiles, created in the previous step, into a network which can be used with NeXTA and DTALite. This process reads the spatial data stored in each specified shapefile, along with the corresponding data stored in the database (DBF) file, to create the input CSV files used by NeXTA and DTALite. In order to interpret the database (DBF) files for conversion, a configuration (INI) file is used to map field names between the shapefiles and the standard data format used by NeXTA.

The network import process is divided into three internal steps:

1. Prepare INI configuration file & attribute files for conversion
2. Use NeXTA's import network tool to convert the network
3. Save the new network as a new project file

1) Prepare INI configuration file & attribute files for conversion

The first step in converting the network is to create a configuration file (*.INI) in the folder containing the exported shapefiles. NeXTA uses this user-defined configuration file (*.INI) to associate/map fields in shapefile DBF files to NeXTA/DTALite's data format, allowing NeXTA to read the network geometry from shapefiles and create a NeXTA/DTALite-compatible transportation network project file.

The configuration file is divided into different sections depending on the type of data to be imported. The first section describes general model attributes and import options. The remaining sections are used to describe the different types of network objects which can be imported. Separate sections are used to import links and nodes, with optional sections for importing zones, zone centroids, and zonal connectors. The Tucson regional model has network coding which is similar to that used by NeXTA/DTALite (i.e., similar attribute names and definitions), simplifying efforts when preparing this conversion configuration file. Detailed descriptions are provided for each critical section below.

Model attributes configuration settings (with reasoning/explanation):

Variable Assignment	**Description/Explanation**
units = MI	The original network describes distances in units of miles.
direction_field =1	"1" indicates that NeXTA uses a "direction" field to interpret the link shapefile.
control_type_field=0	"0" indicates that NeXTA does not read any control type information from the node shapefile.
reserve_direction_field=1	"1" indicates that NeXTA will read direction-specific information for some non-directed links (two-way links).

offset_link=1	“1” indicates that NeXTA will add space between directional links to aid visualization.
link_capacity_flag=0	“0” indicates that the link capacity is described on a per-lane basis.
number_of_lanes_for_two_way_links=0	“0” indicates that lanes on two-way links are described using the number of lanes in each direction.
use_optional_centroid_layer=0	“0” indicates that there is no additional shapefile to read which only contains zone centroids.
use_optional_connector_layer=0	“0” indicates that there is no additional shapefile to read which only contains zonal connector links.

Node import configuration settings (with reasoning/explanation):

Variable Assignment	**Description/Explanation**
reference_file_name = 'Node_WGS84_202ft.shp'	Identifies “Node_WGS84_202ft.shp” as the shapefile which describes the nodes in the network.
node_id = 'ID'	Identifies “ID” as the name of the field which contains unique node numbers.
TAZ = 'TAZID'	Identifies “TAZID” as the name of the field which contains zone numbers mapped to the associated node numbers.

Link import configuration settings (with reasoning/explanation):

Variable Assignment	**Description/Explanation**
reference_file_name = 'Roadway_WGS84_202ft.shp'	Identifies “'Roadway_WGS84_202ft.shp” as the shapefile which describes the links in the network.
from_node_id = 'FROM_NODE'	Identifies “FROM_NODE” as the name of the field which describes the starting node number for one direction on a link.
to_node_id = 'TO_NODE'	Identifies “FROM_NODE” as the name of the field which describes the ending node number for one direction on a link.
link_id = 'ID'	Identifies “ID” as the name of the field which contains unique link numbers.
name = 'ST_NAME'	Identifies “ST_NAME” as the name of the field which contains a street name for each link.
length_in_mile = 'LENGTH'	Identifies “LENGTH” as the name of the field describing the length (in miles) of each link.
direction = 'DIR'	Identifies “DIR” as the name of the field which describes each link’s directional properties (0 or 2 = two-way, 1 = one-way, -1 = opposite direction from FROM->TO notation)
link_type = 'FT'	Identifies “FT” as the name of the field which describes the functional classification (equivalent to link type in NeXTA/DTALite) for the directional link in FROM->TO notation.
number_of_lanes = 'AB_LANES'	Identifies “AB_LANES” as the name of the field which describes the number of lanes on the directional link in FROM->TO notation.

lane_capacity_in_vhc_per_hour = 'AB_CAPACIT'	Identifies "AB_CAPACIT" as the name of the field which describes the lane capacity on the directional link in FROM-> TO notation.
speed_limit_in_mph = 'AB_SPEED1'	Identifies "AB_SPEED1" as the name of the field which describes the free-flow speed on the directional link in FROM->TO notation.
r_link_type = 'FT'	Identifies "FT" as the name of the field which describes the functional classification (equivalent to link type in NeXTA/ DTALite) for the directional link in TO->FROM notation. There was no separate field for functional class in the reverse direction, so the same functional class was used for both directions.
r_number_of_lanes = 'BA_LANES'	Identifies "BA_LANES" as the name of the field which describes the number of lanes on the directional link in TO->FROM notation.
r_lane_capacity_in_vhc_per_hour = 'BA_CAPACIT'	Identifies "BA_CAPACIT" as the name of the field which describes the lane capacity on the directional link in TO->FROM notation.
r_speed_limit_in_mph = 'BA_SPEED1'	Identifies "BA_SPEED1" as the name of the field which describes the free-flow speed on the directional link in TO->FROM notation.

2) Use NeXTA's import network tool to convert the network

The next step is to import the network data into NeXTA. Starting with a new, empty network project in NeXTA, the conversion process is initiated by selecting the Transportation Network Data Set tool in the Import Menu.

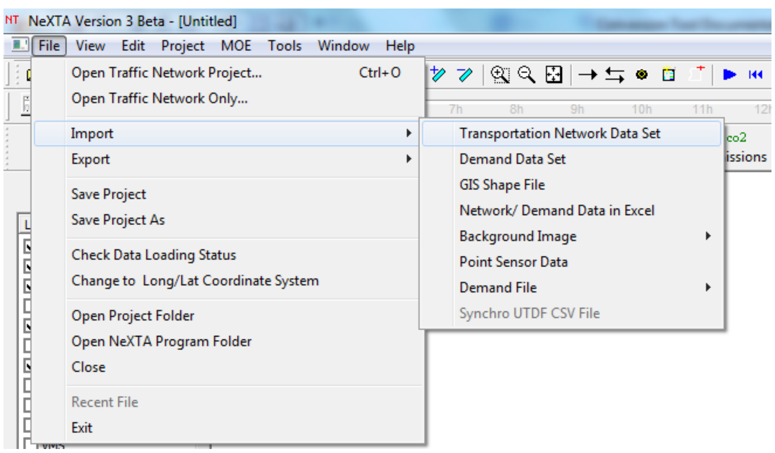

Figure 17: Network Import Menu location in NeXTA

Select the appropriate INI configuration file created in the previous step, and click "Open".

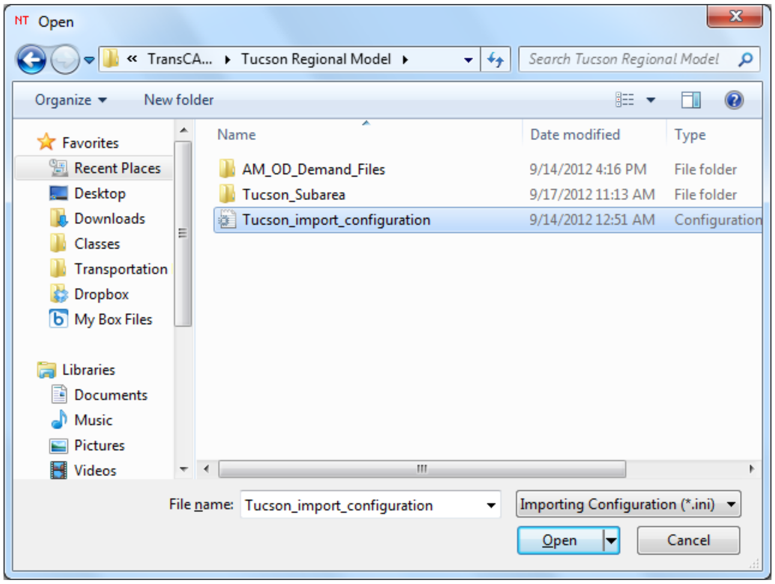

Figure 18: Opening the import configuration INI file in NeXTA

After the conversion process is complete, a File Loading Status window opens to display the results of the conversion process. Make sure to note the number of links, nodes, and zones (if any were selected for import) against the number of records in the shapefiles to make sure the process was completed successfully (no missing records).

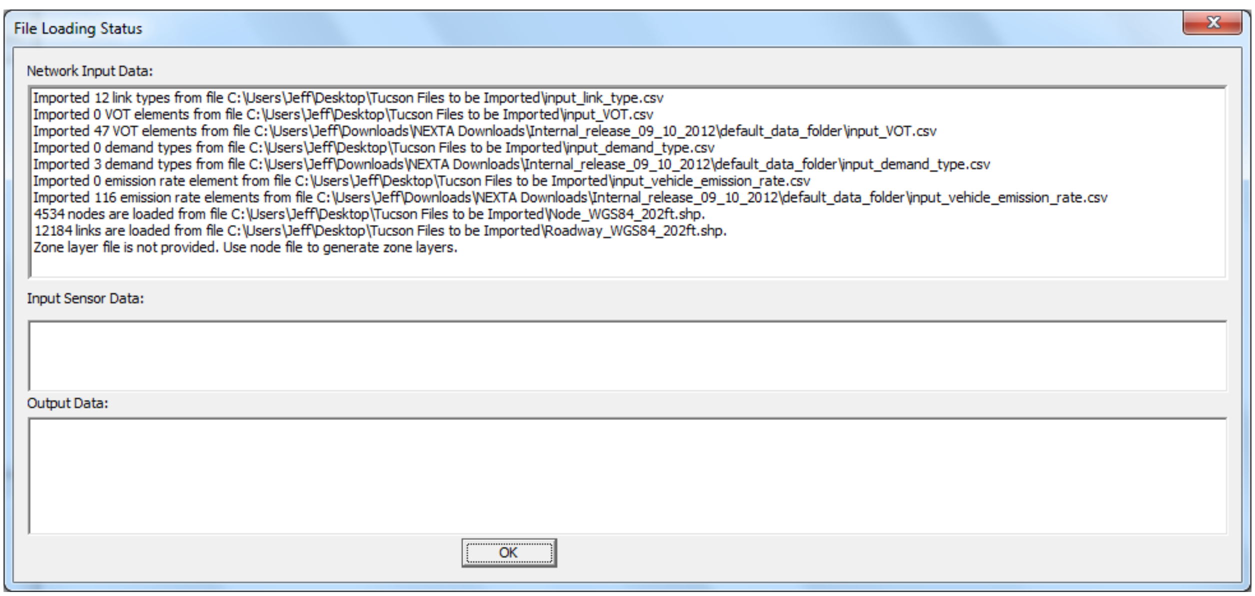

Figure 19: File loading status window displays import results after completion

Notice that NeXTA only imported 12184 links in this network, rather than the 12230 links which are known to exist in the network. This indicates that there are a number of coding issues in the link data which require correction. While some duplicate links contributed to this issue, the primary source of the missing links in this case were several nodes which were added to the network in the originating software application, but the link FROM->TO node numbers were not updated. To resolve this problem, the user must either edit the network in the originating software application, or edit the shapefiles directly in a third-party GIS software application. In this example, detecting and correcting errors in the shapefiles in ArcGIS was completed in approximately one hour.

The final imported network should look similar to Figure 20 below.

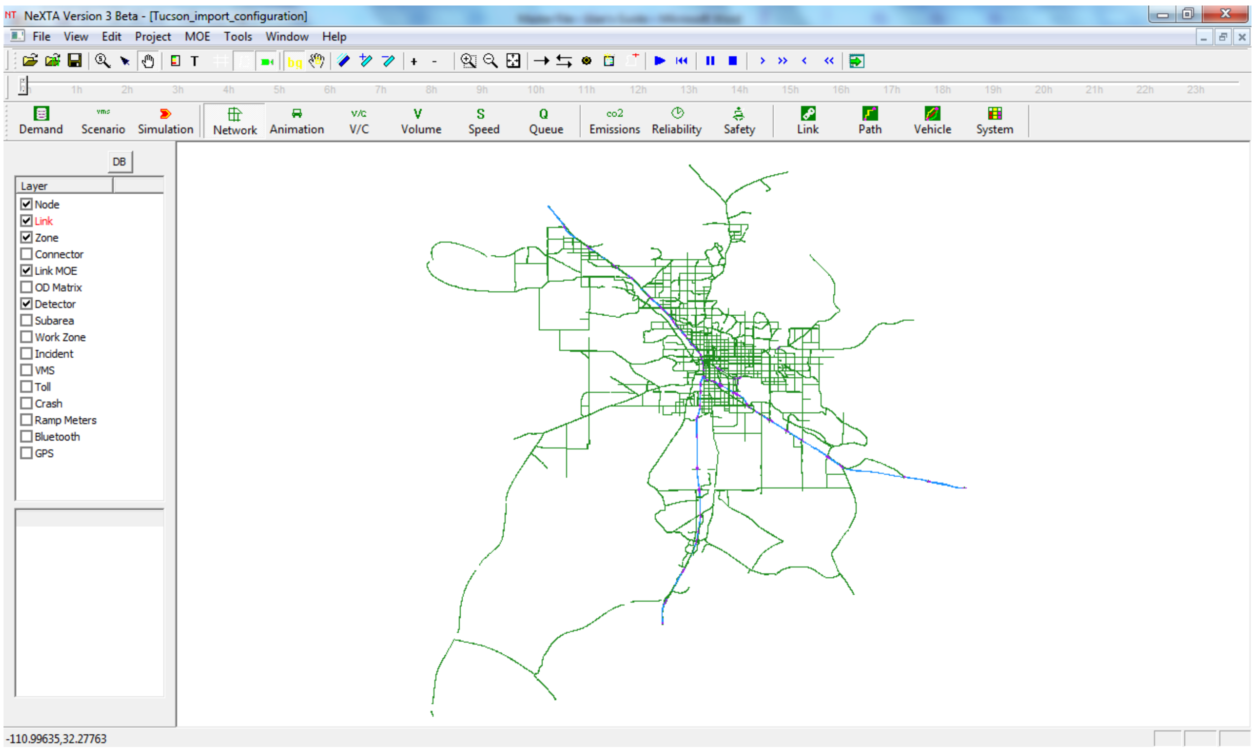

Figure 20: Tucson regional network imported into NeXTA/DTALite

<u>3) Save the new network as a new project file</u>

Finally, save the network as a new Transportation Network Project (*.tnp) file to write the input tables in the project folder.

Step 3: Read demand data into NeXTA from regional travel demand model

The new demand data handling in NeXTA/DTALite has significantly reduced the complexity involved in converting demand data between different software applications. In the current implementation, individual demand tables are read by NeXTA/DTALite when they are specified in the input_demand_meta_data.csv file. Thus, the user must only edit the demand meta-data table to read the CSV files exported from TransCAD for the Tucson regional travel demand model.

This can be accomplished using a simple 5 step procedure for each demand data file to be used in the simulation, as described below with examples in parentheses for the Tucson regional network.

1. Specify the demand file name (e.g., DA_GP.csv) and format (e.g., matrix)

2. Specify the number of lines in the demand file to be skipped by DTALite (zero for matrix)
3. Indicate whether subtotals are present in the last column (zero for none)
4. Specify the loading start time and end time for the demand file (420 to 540, or 7AM to 9AM)
5. Specify the demand types associated with the demand file (only demand_type1)

Step 4: Run assignment with DTALite to equilibrium

Since the objective of this example is to perform more detailed analyses, a subarea will be used to create a smaller network which can be used for microsimulation analysis. Accurately representing the origin-destination flow in a subarea requires creating external zones at the subarea boundary. When the subarea is created, the path flows are aggregated at the links which cross the subarea boundaries, assigning the aggregated trips to the external zone created at the end of those links. However, no path flow data exists without first running the full network simulation. As a result, the first step in creating a subarea network is to create the network path flows by running the DTALite assignment engine to equilibrium.

Now that the network and demand are imported into a format which is compatible with NeXTA/DTALite, running the DTALite assignment engine only requires editing the simulation settings in the input_scenario_settings.csv file, and initiating the assignment engine. To run the assignment engine in NeXTA/DTALite, select Simulation from the Toolbar at the top of the screen.

For this example, the simulation is run for 10 iterations with 310,000 agents, requiring 5 minutes 22 seconds computational time on an Intel Core 2 Duo T7500 (2.2 GHz) with 3GB RAM. Assignment resulted in an average travel time of 14.67 minutes with an average trip length of 7.55 miles.

Step 5: Cut a subarea within the larger model for more detailed analysis

Performing more detailed analyses with large networks like the Tucson regional network often requires creating a subarea network. A subarea network is a self-contained sub-network defined by a subarea boundary, where objects outside the subarea boundary are cut from the network, and new external zones are created at the subarea boundary where links cross the boundary. These external zones are necessary to preserve the origin-destination traffic flow through the subarea.

The subarea creation process is divided into four internal steps:

1. Create a subarea boundary in NeXTA
2. Use NeXTA's Subarea Cut tool to clip the network
3. Convert zonal connectors to side-streets within the subarea
4. Save the new subarea network as a new project file

1) Create a subarea boundary in NeXTA

In order to cut a subarea, a subarea boundary must first be defined. Using the **Create Subarea** tool in NeXTA, a subarea boundary was drawn in the immediate area surrounding the I-10 construction area used for this test case. After drawing the boundary, the links and nodes within the boundary are highlighted by NeXTA, as shown in the figure below.

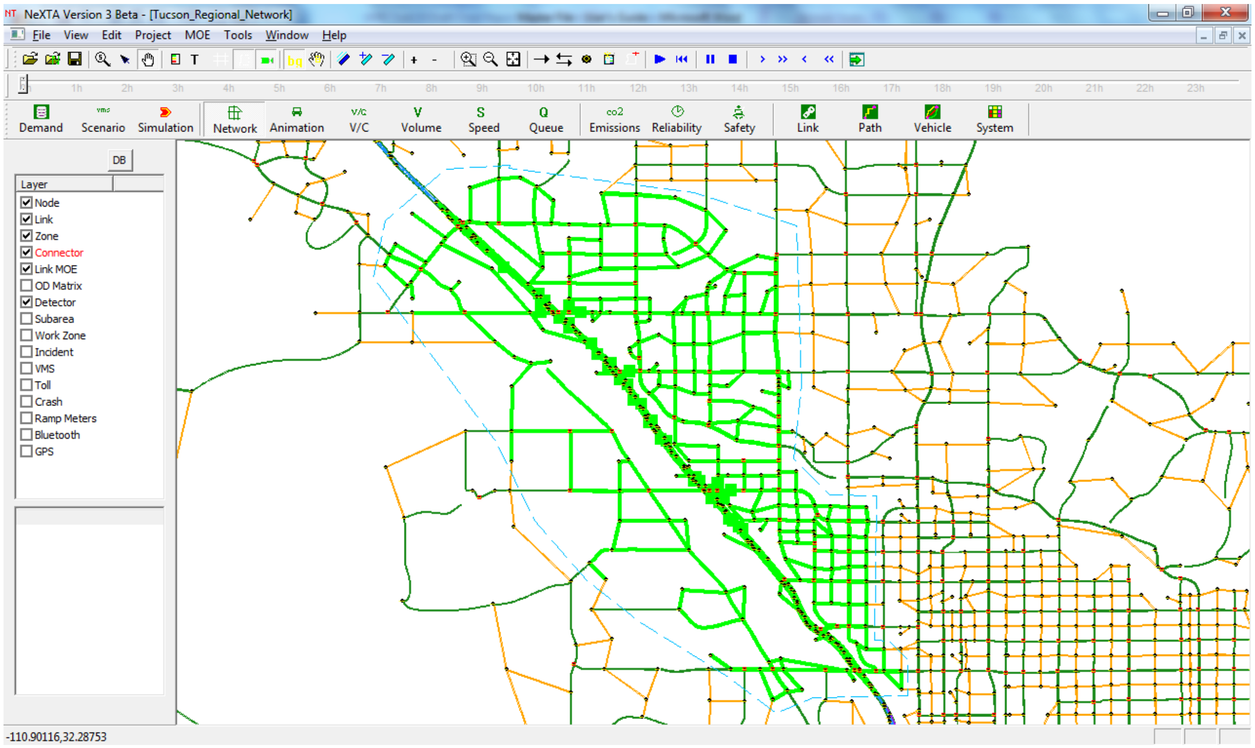

Figure 21: Subarea boundary selection for Tucson I-10 construction area

2) Use NeXTA's Subarea Cut tool to clip the network

Next, use the **Subarea Cut** tool in NeXTA to delete all of the network objects (nodes and links) outside of the subarea boundary by going to **Tools -> Subarea Cut**. NeXTA will display a dialog showing the number of links, nodes, zones, OD pairs, and subarea path records are extracted for the subarea analysis. The resulting subarea network should look similar to the figure below.

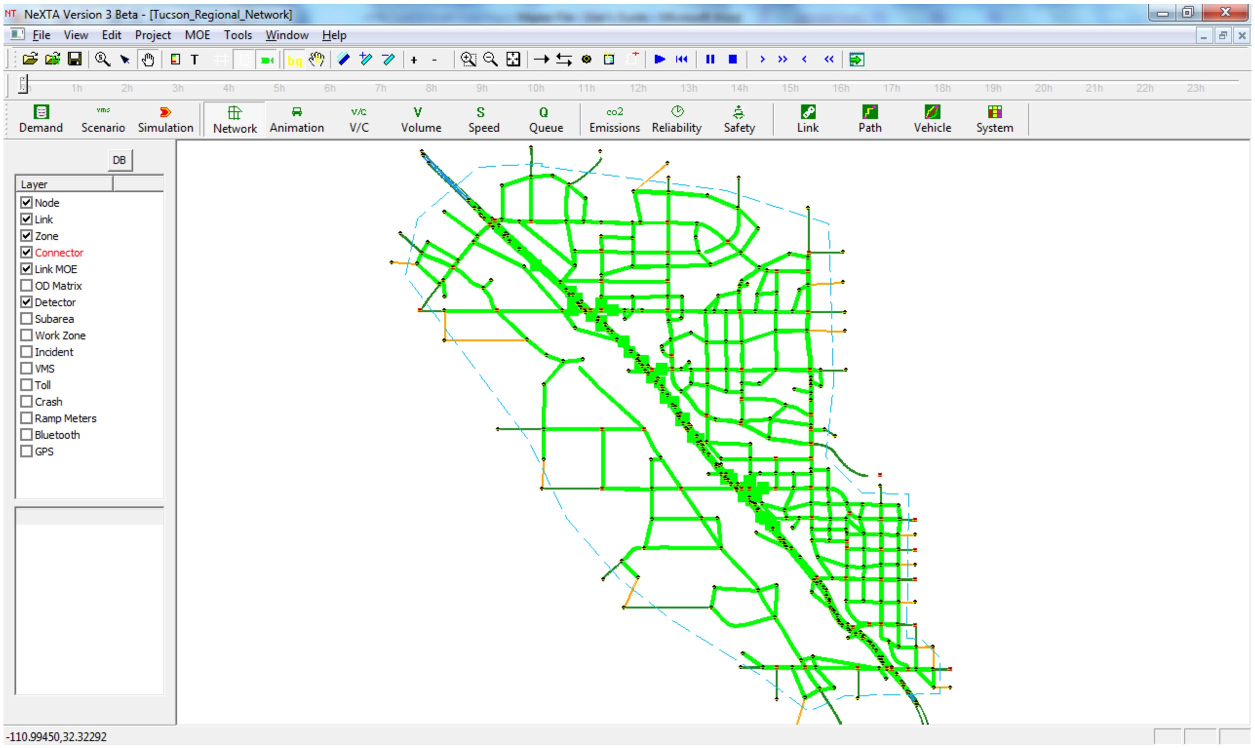

Figure 22: Clipped subarea for the Tucson I-10 construction area

3) Convert zonal connectors to side-streets within the subarea

Lastly, use the **Generate Physical Zone Centroids on Road Network** tool to convert the zonal connectors to side streets within the network by going to **Tools > Generate Physical Zone Centroids on Road Network**. This tool replaces zone centroids with additional nodes so that no paths can be routed through a zone centroid. While DTALite cannot use paths through zone centroids, Synchro and VISSIM can use those paths, making using this tool a very important step in the process toward creating a new Synchro and/or VISSIM network. The resulting network created by using this tool should look similar to the network shown in the figure below.

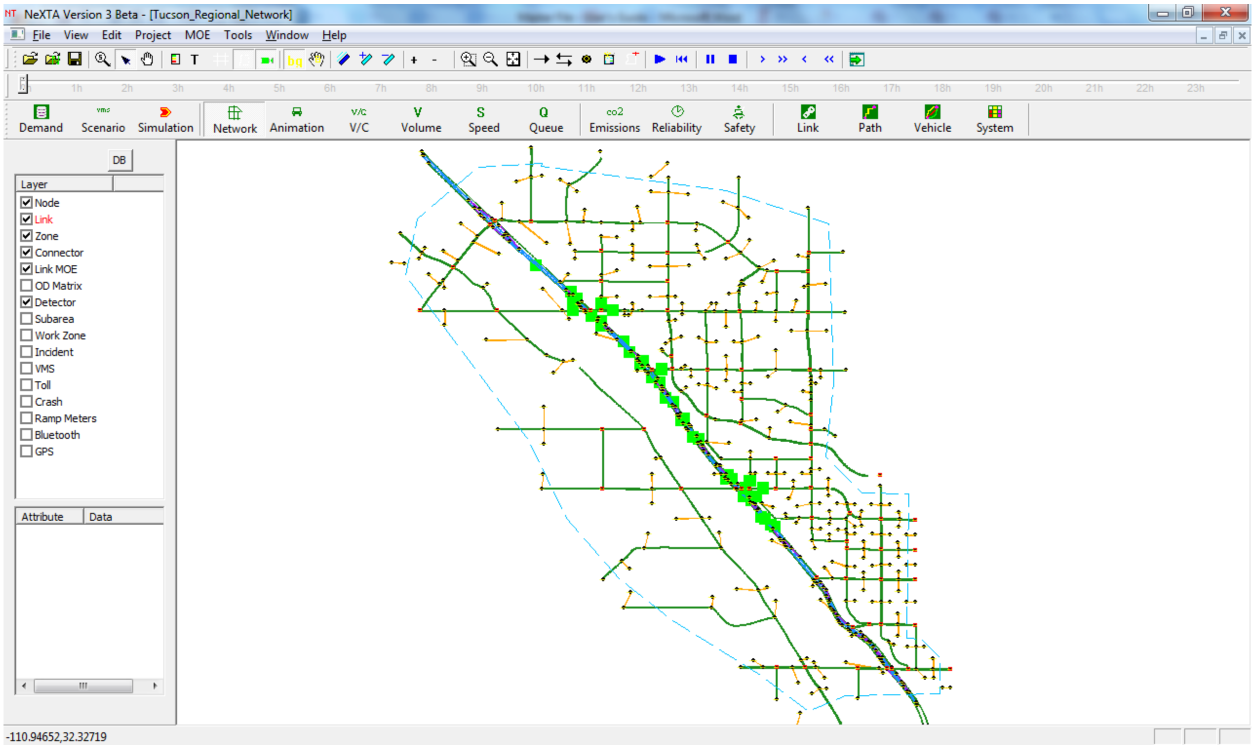

Figure 23: Results from converting zonal connectors to side-streets in the Tucson I-10 subarea

<u>4) Save the new subarea network as a new project file</u>

The last step is to save the new subarea network in a new project file. Go to **File > Save Project As**, and select a new location for the new subarea project TNP file.

Step 6: Prepare field data for ODME (Origin-Destination Matrix Estimation)

Origin-Destination Matrix Estimation is a technique used to adjust demand patterns in a network to better approximate observed traffic conditions (e.g., time-dependent link volume). This is normally accomplished in an iterative process by assigning trips to paths in a network, comparing observed and simulated link volumes/counts, adjusting the input demand data, and moving to the next iteration where the trips are re-assigned. However, before running ODME, the field data must be prepared in a format which enables the ODME model to compare observed and simulated link volumes/counts.

DTALite's ODME model reads field data from the input_sensor.csv file, which the user must prepare before executing the ODME process. This input file uses a flexible format for reading multiple types of observed data in the network, including link volume, occupancy, speed, and travel time field data for specific locations and time periods, allowing for time-dependent ODME applications. The Tucson I-10 subarea field data was prepared from link volume counts collected at 37 locations on freeways and arterials in the subarea model, with hourly and 15-minute link volume counts. Their locations are represented as green squares in the figure below.

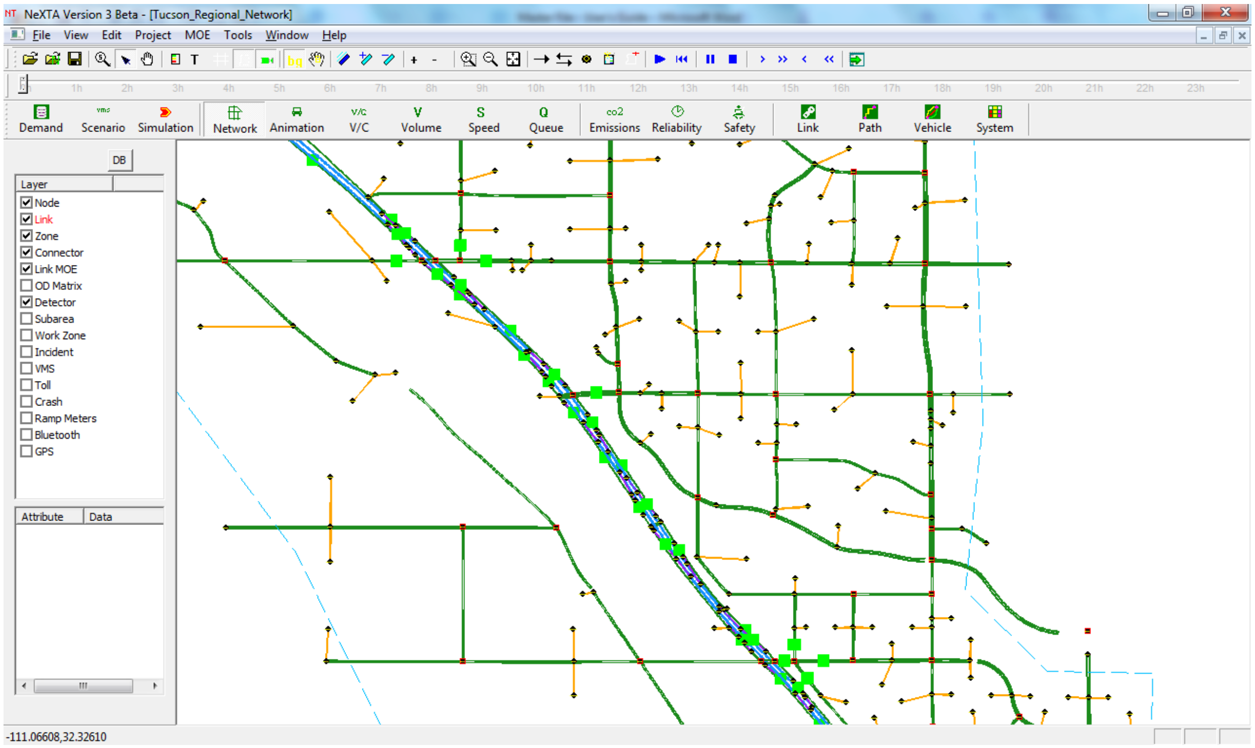

Figure 24: Subarea field data sensor locations for ODME in DTALite

Step 7: Run ODME using field data for calibration

This step-by-step example describes a process for delivering the network geometry and demand data, represented by OD path flows, to be used in microsimulation analysis. Since microscopic simulation relies upon detailed, accurate input demand data to best represent observed traffic conditions in the analysis subarea, it is important to make sure that the OD path flows closely align with these observed traffic conditions. This is accomplished using the ODME component in DTALite, which is used similarly to a network calibration tool in this procedure.

To enable ODME mode in DTALite, the user simply changes a flag variable in the input_scenario_settings.csv file, and adjusts the model parameters in the ODME_Settings.txt file. For this example, the assignment engine is specified to run for five iterations to arrive at a relatively stable operating condition, and then uses the ODME module to adjust path flows over the following ten iterations. The ODME module in DTALite was set to adjust 15% of the OD demand at each iteration, allowing the model to run to completion faster without sacrificing solution quality. Time-dependent link volume/count data was used for calibration, allowing for more detailed OD flow refinements.

The two plots shown below compare the observed and simulated link volumes/counts at the subarea sensor locations. The initial equilibrium assignment (before ODME) produced link volumes that were relatively similar to the observed link volumes with R^2 = 0.74, although under- and over-estimation was observed at multiple locations. After running ODME for 10 iterations, the under- and over-estimation was significantly reduced, and the R^2 improved to 0.89 over all observations.

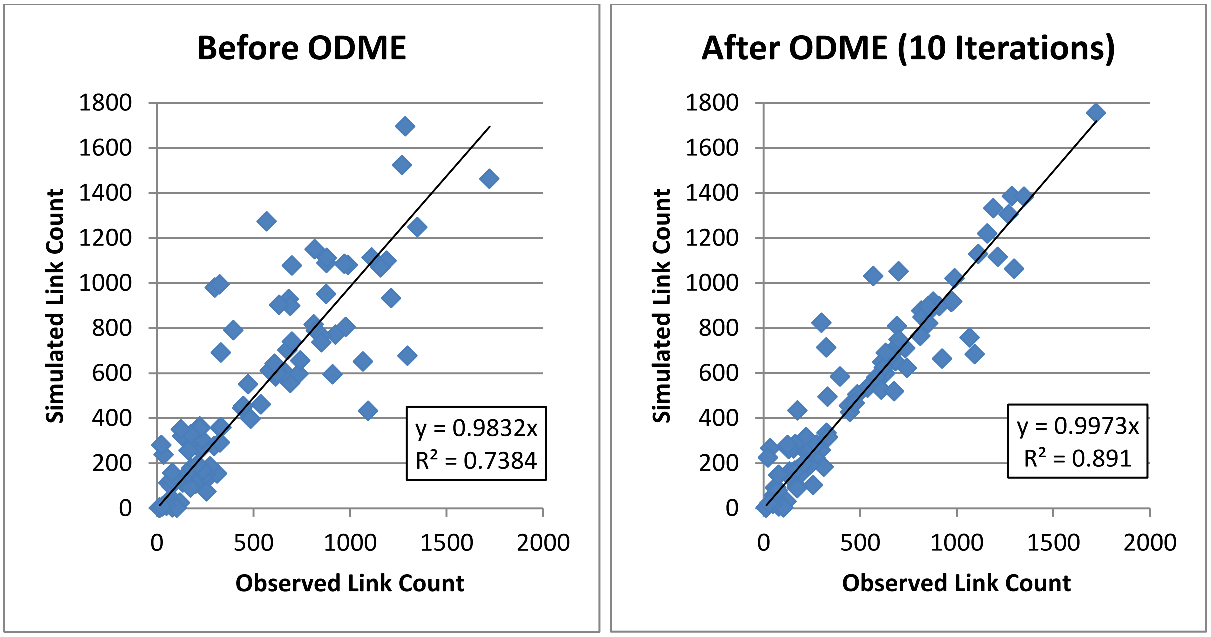

Figure 25: Origin-Destination Matrix Estimation (ODME) results for Tucson subarea

Step 8: Export to Synchro/VISSIM for signal optimization/microscopic analysis

After adjusting the demand data to better represent observed conditions in the subarea network, the subarea network is ready to be exported to Synchro™ and VISSIM™ for microscopic analysis. However, since NeXTA does not currently support importing signal timing data from the regional demand model (if any exists), initial signal phasing and timing is estimated using a Quick Estimation Method (QEM) approach, which is implemented as an automated, external spreadsheet. The procedure for exporting the subarea network for microscopic analysis is divided into three steps:

1. Use QEM to estimate initial signal phasing and timing
2. Export to Synchro using UTDF CSV format
3. Export to VISSIM using ANM format

1) Use QEM to estimate initial signal phasing and timing

An automated QEM spreadsheet is used to generate initial signal phasing and timings for the subarea network, which are exported to Synchro for analysis in the next step. To initiate the QEM spreadsheet calculations, go to **File > Export > Microscopic Network and Traffic Control Data**, and select Perform Quick Estimation Method (QEM) for Signals. NeXTA writes the geometry and volume information to the spreadsheet, the spreadsheet calculates appropriate phasing and timing data, and NeXTA reads that phasing and timing data back into NeXTA.

2) Export to Synchro using UTDF CSV format

Next, the network is exported into UTDF CSV files, which Synchro uses to create the subarea network through its internal import function. The imported network in Synchro is shown in the figure below.

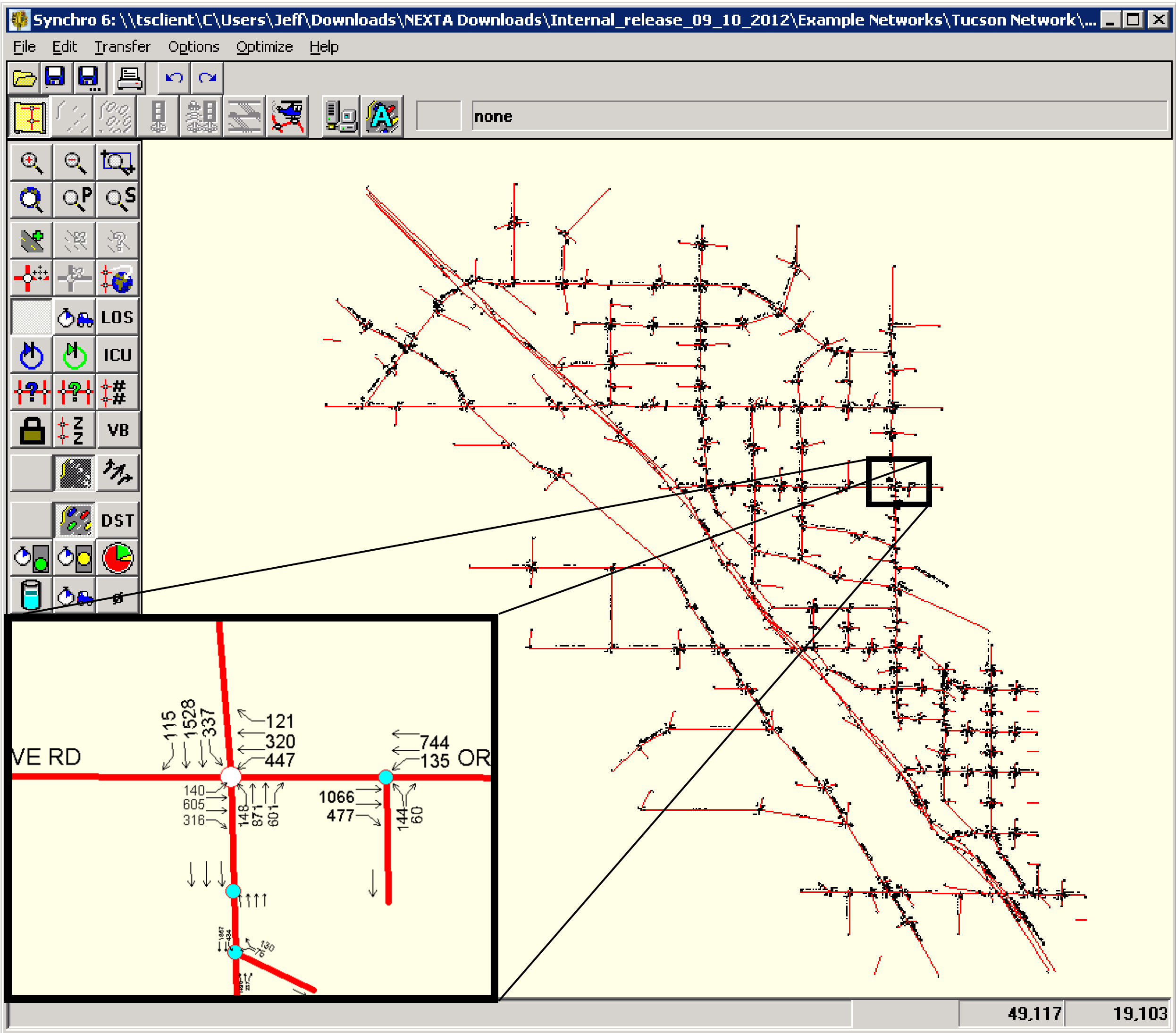

Figure 26: Tucson I-10 subarea network imported in Synchro from NeXTA

3) Export to VISSIM using ANM format

Lastly, the network is exported into the ANM VISSIM format, which VISSIM uses to create the subarea network through its internal import function. The imported network in VISSIM is shown in the figure below.

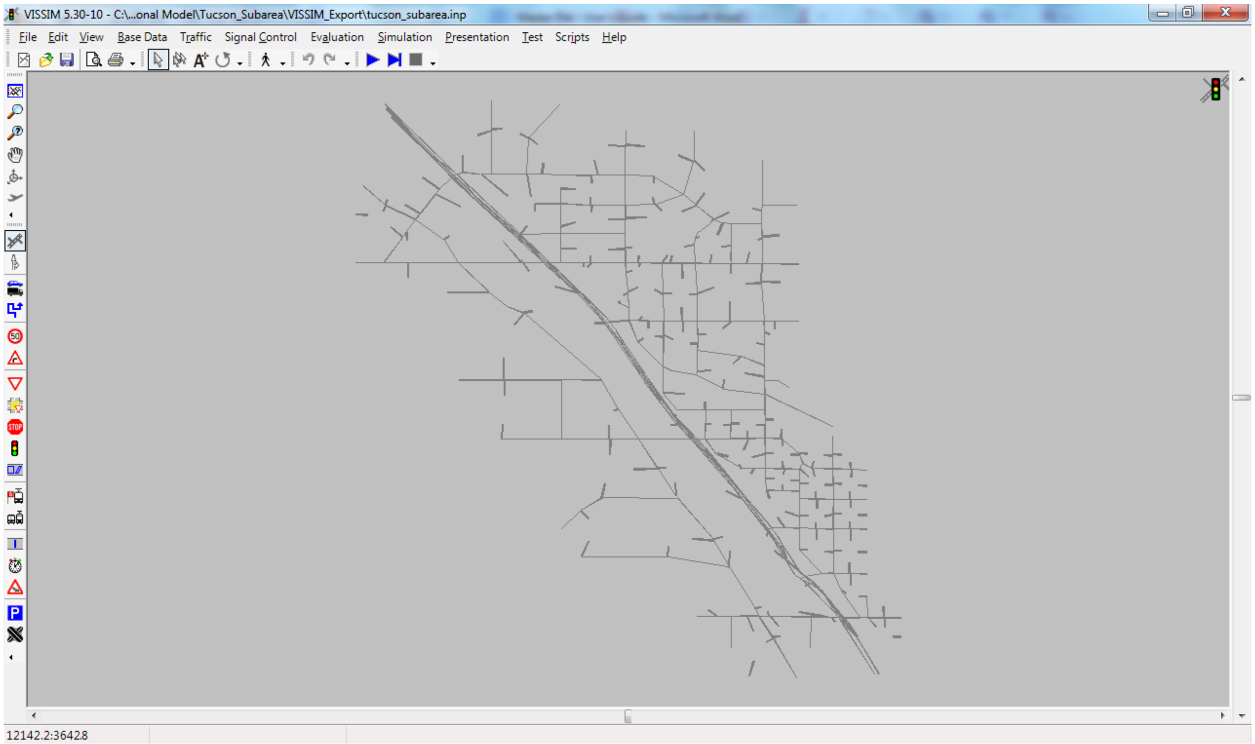

Figure 27: Tucson I-10 subarea network imported in VISISM from NeXTA