# Lesson 7 – Network Analysis

In this lesson we aim to perform network analysis on the OpenStreetMap roads layer. First, some pre-processing must be done to build a network based on our OSM roads layer from the students geodatabase. After building the network, you create an OD-cost matrix, make shortest route calculations, and construct isochrones maps based on the OSM data.

## SETTING UP THE NETWORK FEATURE DATASET

In this lesson, we will use the ArcGIS Network Analysis extension. This allows you to create, edit, and build network datasets.

* Create a feature dataset in which we can store all network-relevant data.

1. Create a feature dataset on the Catalog pane by RIGHT-CLICKING on your geodatabase (\*.gdb) and CLICKING *New  > Feature Dataset *

* Feature datasets are used to facilitate the creation of controller datasets.

1. Type in Network as Feature Dataset Name and execute with Run. (the WGS 84 coordinate system serves us well for now)

## PREPROCESSING: GETTING OSM ROADS READY FOR NETWORK BUILDING

### Reducing Redundant Lines

* It´s beneficial for our network analysis if we reduce the number of redundant lines on our road layer. Make a new road layer without the cycleways and footways (both are parallel to other lines).

1. Build a clause in *Select by Attributes* using the field highway to initially select all cycleway and footway features of your roads layer.
2. Run the *Select by Attributes* tool by inverting the clause to select all other features (except cycleways and footways)
3. Make a new layer from the selected features for further preprocessing in 7.2.2.

### Cleaning Topology Errors

There are different types of topological errors and they can be grouped according to whether the vector feature types are polygons or polylines. A common topological error with **polyline** features is that they do not meet perfectly at a point (node). This type of error is called an **undershoot** (1) if a gap exists between the lines, and an **overshoot (2)** if a line ends beyond the line it should connect to (see [figure](https://docs.qgis.org/2.18/en/docs/gentle_gis_introduction/topology.html#figure-topology-errors)).



The result of overshoot and undershoot errors are so-called ‘dangling nodes’ at the end of the lines. Dangling nodes are acceptable in special cases, for example if they are attached to dead-end streets.

* We aim to correct the topology by reducing the amount of undershooting and overshooting in our road dataset.
* You can find many topology errors in the roads layer. I.e., zoom to the crossroads *Theresienstr. / Arcisstr.* to see one example of an undershoot.

In order to troubleshoot the undershoots, we can use the *Extend Line* tool. It extends line segments to the first intersecting feature within a specified distance. If no intersecting feature is within the specified distance, the line segment will not be extended.

1. Open the *Extend Line* tool from the *Geoprocessing* toolbox and input your selected new roads layer created in 7.2.1. Use 2 meters as Extend Length tolerance and keep the Extend to Extensions option checked.
2. CLICK *Run* to fix (most of) the undershoot errors.

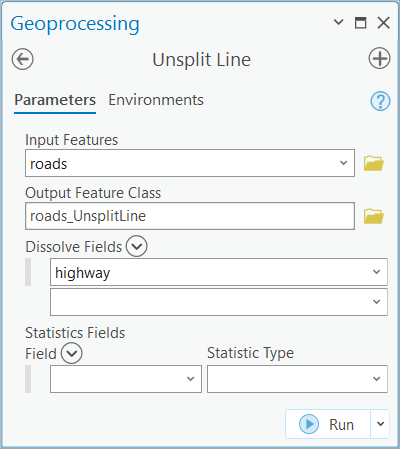
In order to troubleshoot the overshoots, we can use the *Trim Line Tool.* Line segments that are shorter than a specified dangle length and do not touch another line at both endpoints (dangles) will be trimmed.

1. Open the *Trim Line* tool and input your selected new roads layer created in 7.2.1. Use 2 meters as Dangle Length tolerance and keep the Delete Short Features Option checked.
2. CLICK *Run* to fix the overshoot errors.

### Eliminating Pseudo Nodes

Pseudo nodes occur when the end node on a feature connects to another end node with no other nodes present. They often indicate that the two features can be merged. For example, a long road segment may have been divided into two features.

* It´s beneficial for our network analysis if we reduce the number of pseudo nodes on our road layer. But first, we want to make these nodes visible.

1. Open the *Feature Vertices To Points* tool. Use your selected new roads layer as Input. Keep the Point Type option on All vertices.
2. CLICK Run to create a feature class containing points generated from all vertices.
3. View the result. You will be able to identify many pseudo nodes.
4. Open the *Unsplit Line* tool. This aggregates line features with coincident endpoints and, optionally, common attribute values.
5. Use your selected new roads layer as *Input Features*. Define an *Output Feature Class.*
6. Select the field *highway* as *Dissolve Field*. This means features of the same road type (attribute: highway) will be aggregated.
7. Click *Run* to eliminate all pseudo nodes.

### Eliminating Overpasses

We require one final pre-processing step by eliminating overpasses. Overpasses are further topology errors, where one line crosses the other without having a defined node at the crossroads. However, on a network we need to establish turning possibilities at crossroads.

* First, establish nodes at all road intersections. Then, create individual lines between all nodes

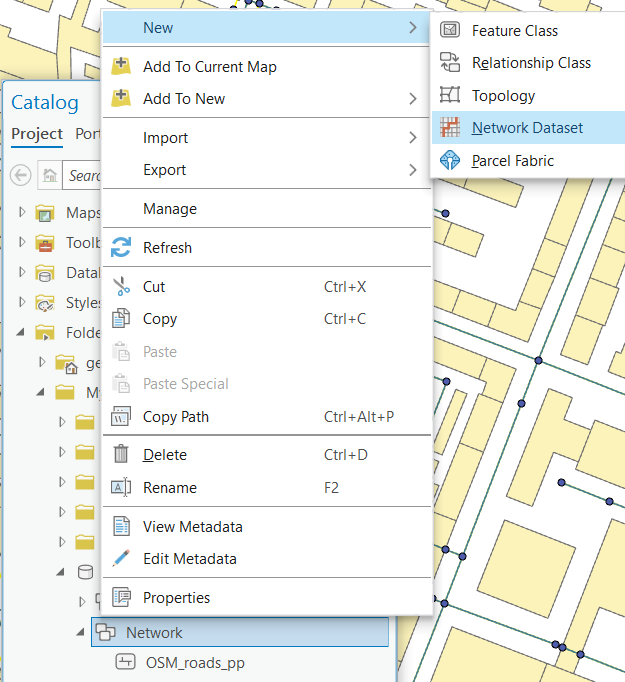
1. Open the Intersect tool. Use your unsplitted line layer as *Input*. Name a *Output Feature Class.*
2. Use *Point* as *Output Type* and *Run*.
3. Open the *Split Line At Point* tool. This splits line features based on intersection or proximity to point features.
4. Use your unsplitted line layer as *Input Features*, the newly created intersected points layer as *Point Features,* and Name the *Output Feature Class* as *OSM\_roads\_pp.* Furthermore, save it into the Network feature dataset created in 7.1. Note that this will be Your final pre-processed layer for building a network.
5. Use 2 meters as the Search radius. This distance will be used to split lines by their proximity to point features. Points within the search distance to an input line will be used to split those lines at the nearest location to the point along the line segment.
6. CLICK *Run* to complete the pre-processing.

## BUILDING A NETWORK LAYER

* Build a network dataset based on the pre-processed *OSM\_roads\_pp* layer

### Preparing a Network Layer

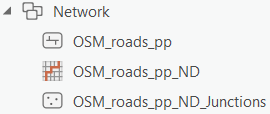
1. Before you start building a network dataset, you need to remove the pre-processed *OSM\_roads\_pp* layer from the *Contents pane*.
2. On the *Catalog* pane, RIGHT-CLICK on your *Network* feature dataset and CLICK *New  > Network Dataset .*



1. Define the *Network Dataset Name* as *OSM\_roads\_pp\_ND.*
2. *Check* to select the *OSM\_roads\_pp* roads layer as *Source Feature Class*.

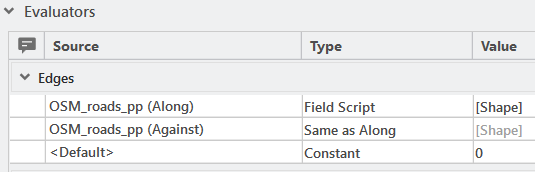
* If the target OSM\_roads\_pp roads layer doesn’t appear in the *Source Feature Class*, RIGHT-CLICK on Network Feature Dataset, CLICK *Import – Feature Class(es)* to import the OSM\_roads\_pp roads layer into the Network Feature Dataset and try again. If still not appear, create a new Geodatabase and a new Network Feature Dataset, and try again.

1. *Elevation model* specifies the model to be used to control vertical connectivity in the network dataset (this would require an attribute filed that defines vertical levels). In this lesson we are keeping it simple. So, select *No elevation*.
2. *Run* the tool*.* Check the *Network* feature dataset on *Catalog.* You should see a new network dataset.



### Setting up a Network Dataset

1. In the Catalog pane, RIGHT-CLICK on *OSM\_roads\_pp\_ND* and open *Properties.* The**Network Dataset Properties** window will open.
2. Under *Built Status*, you can see that the network is not built yet with all topological tables that help enable network analysis.
3. Check the *Service\_area Index* checkbox under *Indexes*. This will let future network algorithms run faster.
4. Switch to the *Travel Attributes* options on the left of the *Network Dataset Properties* window.
5. On the first tab, *Travel Modes*, CLICK on the Hamburger icon and CLICK *New* . This will create a *New Travel Mode.*
6. Staying on this same tab, change the routing type to *Walking* . Leave the *Impedance* on *Length* as cost.
7. Switch to the 2nd tab, *Cost*. You can see that length is now defined as the cost. The length will be calculated based on the geometry, which you can see in the lower part of this tab.



1. On the *Catalog* pane, RIGHT-CLICK on *OSM\_roads\_pp\_ND* and CLICK *Build* . CLICK *Run* to complete the network dataset.

## EXPLORING THE BUILT NETWORK

You can inspect network elements and attributes of a network dataset referenced by a network dataset layer or network analysis layer in the map display using the *Explore Network* tool.

* Check the connectivity of your built network

1. Once You have built the network, this layer is automatically added to the Contents pane. Toggle off all other road layers so You can see the grey-tone Edges of the network layer. CLICK the *OSM\_roads\_pp\_ND* layer on the Contents pane to activate associated functionalities on the ribbon.
2. On the ribbon, CLICK the *Data* tab and then *Explore Network* *.*
3. On the map interface, CLICK on different edges of the network. The selected edge is highlighted, and the adjacent network elements are emphasized. The opening **Explore Network** window shows the cost to traverse the particular edge (*impedance; in our simple case,* the cost is length).

## CREATING AN ORIGIN-DESTINATION COST MATRIX

* Before we start, we create a new point feature class named nodes. These will serve as stops for the OD cost matrix.

1. Create a new point feature class named *nodes* into the *Network* feature dataset*. Check* M Values as geometric properties, but *uncheck* Z Valuesbefore completing the point feature class.
2. Create 4-6 stops over the network via Edit > Create by using the *create a point feature* and clicking on the map interface. The points should be snapped to the network (no need to be fully snapped). Try to distribute Your points over the network. Save the edits.

* Here we will create a distance overview between our points using Djkstra´s shortest path algorithm on our built network.

1. On the Contents pane, select the layer *OSM\_roads\_pp\_ND* by a single CLICK.
2. On the top ribbon, go to Analysis > Network Analysis . You should see that our layer is pre-selected as network data source.
3. CLICK *Origin-Destination Cost Matrix* . You will receive a new layer group named OD Cost Matrix on the Contents pane.
4. CLICK on *Origins*. Then, on the ribbon, go to the *OD Cost Matrix Layer* to see all tools.
5. CLICK on *Import Origins.* To Add Locations, s*elect* your stop layer *nodes* as *Input Locations and CLICK Apply.*
6. Repeat the same procedure for *Destinations* with the same *nodes.*
7. Use the *New Travel Mode* (under Travel Settings - Mode) and Straight Lines options (under Output Geometry) and CLICK *Run* to execute the *Origin-Destination Cost Matrix.*
8. CLICK and Label the new *Lines* layerwith the attribute *total meters (Total\_Length)* by using the options on the *Labeling* tab*.* You should be able to see the labelled distance values calculated along the network.
9. *Optionally, you can change the primary symbology of the lines* to *Proportional Symbols* to view the differences

## SHORTEST ROUTE CALCULATION

* Here we will create a shortest path between two arbitrary points using Djkstra´s shortest path algorithm on the OSM network.

1. On the Contents pane, select the layer *OSM\_roads\_pp\_ND* by a single CLICK.
2. On the top ribbon, go to *Analysis > Network Analysis >* Route . You will receive a new layer group named *Route* on the Contents pane. CLICK on the new *Stops* layer.
3. On the *Route Layer* tab (on ribbon), CLICK on the *Create Features* button next to *Import Stops* button to add two arbitrary *Stops* on the network*.*
4. Under *Output Geometry*, use the *Along Network* option and CLICK *Run* to execute the *Route calculation.*
5. Inspect the result by clicking on the newly drawn route. You can find the shortest distance under the attribute *Total\_Length*.

## TACKLING THE TRAVELLING SALESMAN PROBLEM

To tackle the travelling salesman problem, a GIS simply reorders all stops to find the shortest possible route. A routing algorithm (here: Dijkstra) is applied to every possible combination of ordered visits. Therefore, this procedure becomes very CPU-intensive for higher number of stops on a larger network.

* Find the shortest route visiting a number of stops when the order does not matter.

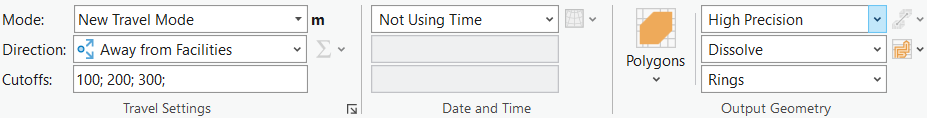
1. CLICK on the *Stops* layer on the *Contents* pane.
2. On the *Route Layer* tab CLICK on the *Create Features* button next to *Import Stops* button to add two arbitrary *Stops* on the network*.* Alternatively, You can CLICK on *Import Stops* button, s*elect* your stop layer *nodes* as *Input Locations.*
3. Under *Travel Settings,* choose the *Sequence Find Best.* *.*
4. Under *Output Geometry*, use the *Along Network* option and CLICK *Run* to execute the *Route calculation.*
5. Inspect the result by clicking on the newly drawn route.

## CONSTRUCTING ISOCHRONES

Isochrones are lines drawn on a map connecting points which are accessible at the same time. In ArcGIS, you can use the Network Analyst to construct isochrones

* Construct isochrones based on public transport stops.

1. On the Contents pane, select the layer OSM\_roads\_pp\_ND by a single CLICK.
2. On the top ribbon, go to Analysis > Network Analysis > Service Area  . You will receive a new layer group named *Service Area* on the Contents pane. CLICK on the new *Facilities* layer.
3. On the *Service Area Layer* tab (on ribbon), CLICK on *Import Facilities.* To Add Locations, *select* a public transport stop layer (such as underground, tram or bus stop point layer from previous lessons) as *Input Locations and CLICK OK.*
4. Under *Travel Settings*, define reasonable *Cutoffs*. They are in meters.
5. Set the *Output Geometry* options as you like.
6. CLICK *Run* to execute the Isochrone calculation*.* View the result.



Contents

[Lesson 7 – Network Analysis 1](#_Toc180590755)

[7.1 SETTING UP THE NETWORK FEATURE DATASET 1](#_Toc180590756)

[7.2 PREPROCESSING: GETTING OSM ROADS READY FOR NETWORK BUILDING 1](#_Toc180590757)

[7.2.1 Reducing Redundant Lines 1](#_Toc180590758)

[7.2.2 Cleaning Topology Errors 2](#_Toc180590759)

[7.2.3 Eliminating Pseudo Nodes 3](#_Toc180590760)

[7.2.4 Eliminating Overpasses 4](#_Toc180590761)

[7.3 BUILDING A NETWORK LAYER 5](#_Toc180590762)

[7.3.1 Preparing a Network Layer 5](#_Toc180590763)

[7.3.2 Setting up a Network Dataset 6](#_Toc180590764)

[7.4 EXPLORING THE BUILT NETWORK 7](#_Toc180590765)

[7.5 CREATING AN ORIGIN-DESTINATION COST MATRIX 7](#_Toc180590766)

[7.6 SHORTEST ROUTE CALCULATION 8](#_Toc180590767)

[7.7 TACKLING THE TRAVELLING SALESMAN PROBLEM 8](#_Toc180590768)

[7.8 CONSTRUCTING ISOCHRONES 9](#_Toc180590769)