NS Lab 4 - Routing

Junchao Wang (j.wang2@uva.nl)
Rex Valkering (r.a.valkering@uva.nl)
Daniël Louwrink (d.louwrink@uva.nl)
Daan Kruis (d.kruis@uva.nl)

Lab date: Sept. 23 and Sep.28, 2016
Hand-in time (submit to blackboard) by Sept. 29 @23:59
Total points: 20

Abstract

This assignment focuses on understanding how packets route from a source to a destination through a number of routers. You will learn how to manipulate an Internet topology with NetworkX and how to routing between different regions using shortest path.

This lab must be done individually.

Preparation

For this assignment you must use python (suggested version 2.7.X, because there can be big differences between python 2.X and 3.X).

Task 0 – Installation

NetworkX is a nice and handy tool for handling graphs. You can use the NetworkX to manage the graph.

You can find the installation instructions of Network at:

http://networkx.readthedocs.io/en/networkx-1.11/install.html

Internet Topology Zoo is a "Zoo" collecting real network topologies around the world. They not only offer data set, but also tools for parsing, editing and visualizing the data. You can find the dataset we are going to use for this lab through:

http://topology-zoo.org/dataset.html

In this lab we are going to use the **'aarnet.gml'**.

Task 1 – Hands-on on real Internet topo (5 pts)

Write a short Python program that reads the zoo topology; you can use the APIs provided by NetworkX to load the file.

Read the topology and transform it into a numbered graph. The number for each node in the topology should follow the alphabetical order of the nodes in the topology. For example, if in a graph there are three places: ”Amsterdam, Eindhoven, Utrecht”, then “Amsterdam” corresponds to 0; “Eindhoven” corresponds to 1 and “Utrecht” corresponds to 2 (Notice: The starting number is 0).

Produce a plot (png file) of the topology.

Submissions:

Submit the source code (naming: ‘YourFirstName-YourLastname-transform.py’) and the plotted graph(naming: ‘YourFirstName-YourLastName-transform.png’).

Requirements:

1. You can use the ‘convert_node_labels_to_integers‘ function provided by NetworkX to verify your result, but you cannot use it in this assignment.
2. You should use the pydot to generate the dot file and then transform it into a ‘png’ or ‘pdf’.

Tips:

1. You can use “read_gml” which is an API provided by networkx to load the topology data.

Task 2 –Shortest path between two nodes (5 pts)

In networkx, there is a tool for calculating the shortest path between a source and destination, but for this assignment you cannot use it.

Use the Vincenty distance (https://en.wikipedia.org/wiki/Vincenty%27s_formulae) as the cost between the nodes.

Tips: You can use geopy to calculate the Vincenty distance.

Requirements:

1. Calculate the shortest distance between ‘Adelaide1’ and ‘Brisbane1’ and list the number of the nodes on the path.
2. List the name of the intermediate nodes on the path.
3. Plot the shortest path with red color on the graph.
4. Provide the source code (YourFirstName-YourLastname -distance.py) and the plotted graph (YourFirstName-YourLastname -shortestpath.png).

Task 3 –Calculate the shortest path between all pairs of nodes (5 pts)

Use the previous function to calculate the shortest path between all pairs of nodes. You should not directly use the functions provided by NetworkX to calculate the shortest path. Then calculate the diameter of the network (Longest path of the network).

Requirements:

1. Calculate the diameter of the network
2. Write down the nodes in the path with length of diameter.
3. Assume the distance between 1 and 6 with shortest path can be represented as distance (1,6). How many pairs of nodes with distance that exceed the distance (1,6)?
4. Provide the source code (YourFirstName-YourLastname -alldistance.py) and write down the answer in answer sheet.

Task 4 –QoS-aware routing (5 pts)

There can be unexpected failures between the links. Therefore, the network should tolerate some link failures, which means that a link fails, the network should still be able to work properly.

Requirements:

1. Assume that the link between 12 and 17 encounters some problem, does it affect the diameter of the network?

2. Assume the network can tolerate 2 link failures, how many possible combinations that can incur the change of the diameter?

Submission:

In essence you submit:

1. Code: YourFirstName-YourLastname -transform.py, YourFirstName-YourLastname -distance.py, YourFirstName-YourLastname-alldistance.py
2. Plots: YourFirstName-YourLastName-transform.png , YourFirstName-YourLastname -shortestpath.png

2. Answer sheet: YourFirstName-YourLastName.pdf with answers to the questions (including plots).