Performance Evaluation – Master UBINET
Assignment 1

Solutions have to be sent by January 4th 2014 to `giovanni.neglia@inria.fr`.

Ex. 1 — (Network analysis)
Each of you has assigned a different undirected graph whose nodes have coordinates. For each graph there are two files: one contains the edge list (Nedgelist.txt, a line "n m " denotes the link (n, m)) and the other the coordinates of each node (Nposition.txt, a line "n : x y" denotes that node n has coordinates (x, y)). There is also a smaller graph you can use to test your code (test_edgelist.txt, test_position.txt). The name of your graph is indicated in the table below and all the files are available at `www-sop.inria.fr/members/Giovanni.Neglia/perf13/`. Let $\mathcal{G}$ denote the original graph, you have to generate two other graphs $\mathcal{G}'$ and $\mathcal{G}''$ according to the following rules:

- for $\mathcal{G}'$: for each node in G add a link to another node selected uniformly at random;
- for $\mathcal{G}''$: for each node n in G with coordinates $\vec{\mathbf{x}}_\mathbf{n}$ add a link connecting n to the node closest to the target $\vec{\mathbf{x}}_\mathbf{n}+\vec{\mathbf{u}}_\mathbf{n}$, where $\vec{\mathbf{u}}_\mathbf{n}$ is a random vector generated as follows: its angle is uniformly distributed in $[0, 2\pi]$ and its length is a continuous random variable with probability density $2/d^3$ for $d \in [1, \infty)$ and zero otherwise.

For each of the three graphs calculate the diameter, the clustering coefficient and the average length (in terms of number of hops) of the shortest paths and plot the degree distribution. Comment the results.
Consider a greedy routing algorithm where at each node a message is forwarded to the neighbor closest to the destination and evaluate its performance on the three graphs. Does the algorithm guarantee that each message is delivered? For the messages that the greedy routing algorithm can deliver successfully calculate the average length (in terms of number of hops) of the paths identified. Compare it with the average length of the shortest paths.
You can develop from scratch your own code (in C, Java, Python or MatLab), but you are strongly invited to use NetworkX library (`http://networkx.lanl.gov/`).

Your result have to be presented in a report, describing the methodology, your code and the results you obtained. Partial answers, like bounds or Monte Carlo estimations for the quantities of interest, will be taken into account for the final mark.

Student	Graph number
Bernards	1
Carunchio	2
Kaddouri	3
Soni	4
Soroush Haddadi	5
Tetley	6
Toth	7
Varava	8
Zhang	9
Zholtkevych	10

Table 1: Student-trace matching.