Magno Alessandro: 4478234

First assignment report

The real network I used to perform the task comes from a peer-to-peer file sharing nework, named Gnutella peer-to-peer network, collected on august 2002.

Dataset information

The network is a directed graph generated from a sequence of 9 snapshots of an unspecified peer-to-peer network, which use the Gnutella protocol. The nodes represent hosts in the Gnutella network topology and edges represent connections between the Gnutella hosts to send and receive data.

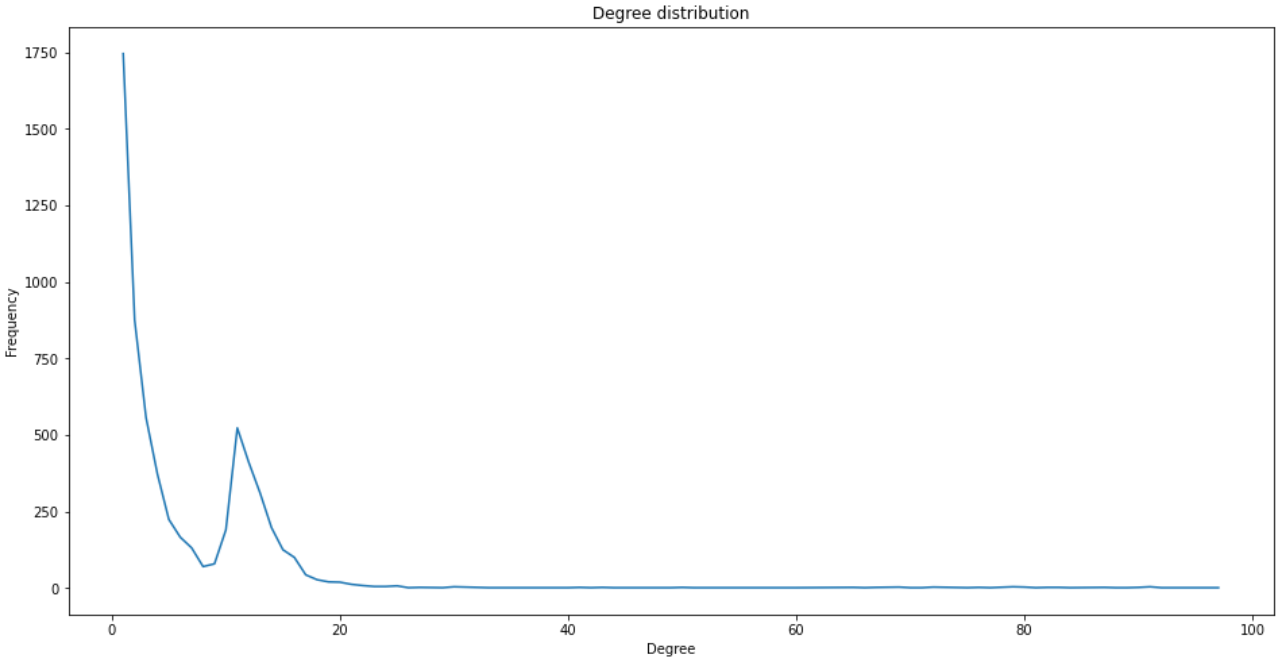
Metrics

Metric	Value
Number of nodes	6301
Number of edges	20777
Density	0.0005
Average Degree	3.297
Average Shortest Path	6.629
Size of Giant Component	6299
Biggest strong connected component size	2068
Global clustering	0.021

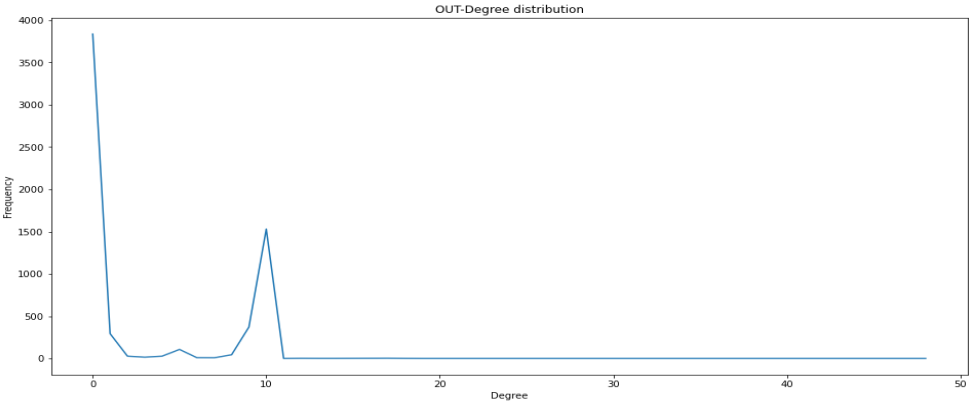
The metrics have been calculated considering the graph as undirected.

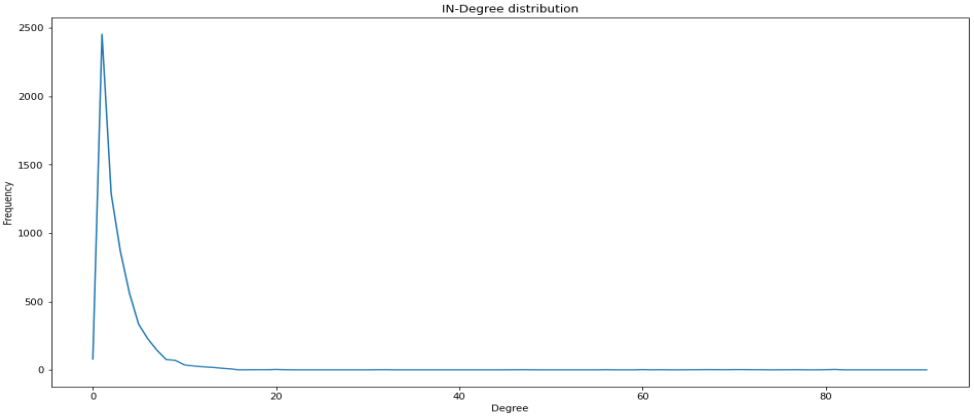
I can notice from them that the network is well connected, in fact it’s almost a unique giant component (6301 total nodes, 6229 nodes of giant component, only 2 nodes are outside) and 1/3 (2068/6301) of the network is string connected. The low density and the global clustering suggests that the network is sparse.

Degree distribution



The degree distribution built by in-degree and out-degree, seems to follow a strong power low but between 10 and 20 it is present a peak. To understand why there is that peak in the distribution, I plotted also the in/out-degree distribution.



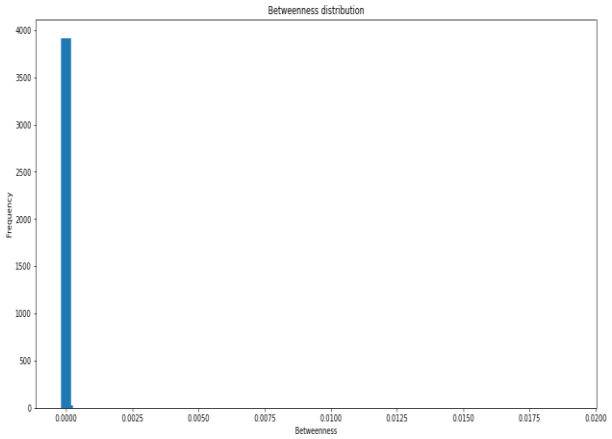


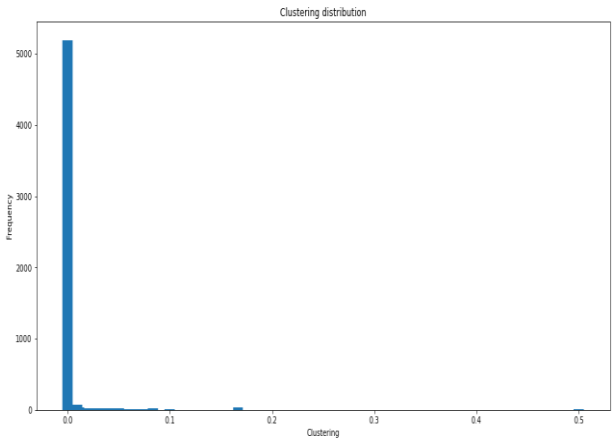
From the plots, I can notice that the in-degree distribution follows the power-law and the out-degrees shows the peak, which means that a huge number of hosts share files with an average of 10 nodes and the hosts who download files struggle to have more connections. I have done some research and the peak can be caused by the algorithm: Gnutella is a p2p protocol and depending on how the algorithm is implemented, the peak could be a wanted/unwanted behavior. It is possible that happened in this case because from 10 to 11 the number of nodes with these connections drop down from more than 1500 to near 0. So it appears that the algorithm put a sort of limit of out connections a host can made.

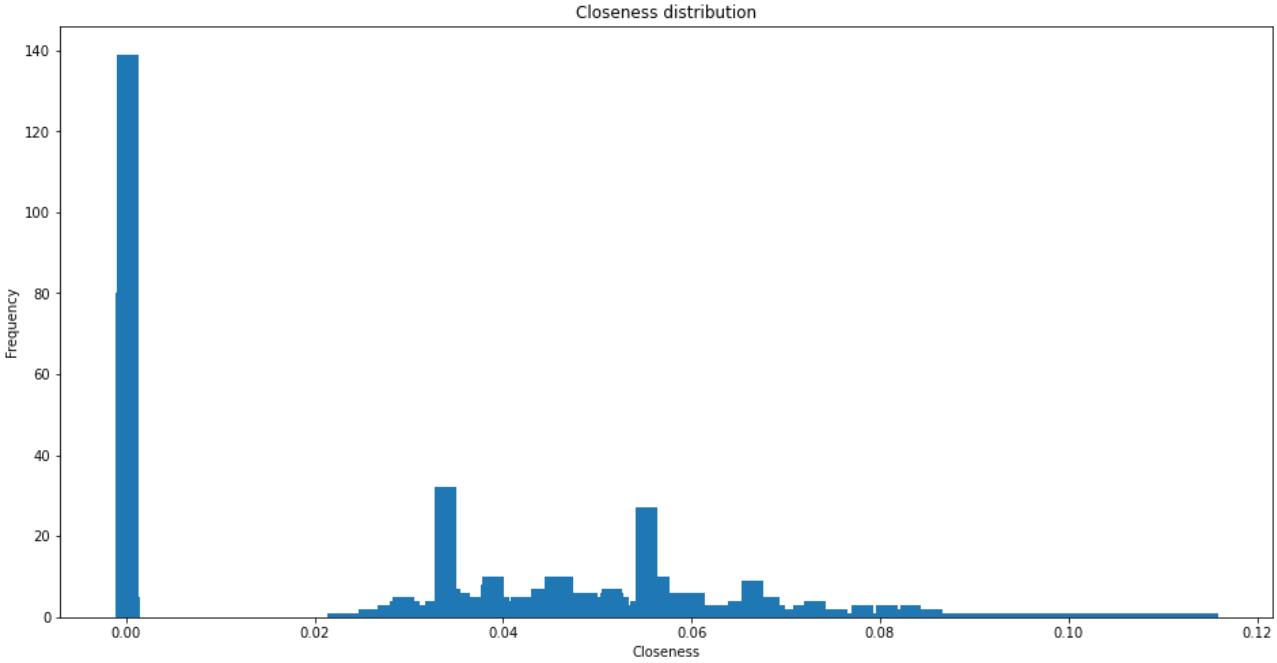
Top 10 nodes by metrics

Degree		Betweenness		Closeness		Clustering	
Node	**Value**	**Node**	**Value**	**Node**	**Value**	**Node**	**Value**
123	97	1317	0.018	367	0.114	506	0.5
127	95	3	0.017	249	0.113	702	0.5
367	94	146	0.014	145	0.111	3589	0.5
424	92	390	0.014	264	0.111	4223	0.5
264	91	175	0.014	266	0.110	4278	0.5
251	91	559	0.011	123	0.110	4321	0.5
427	91	1534	0.011	427	0.110	5060	0.5
266	91	250	0.011	127	0.110	6287	0.5
249	90	700	0.010	122	0.109	4022	0.33
145	90	264	0.010	5	0.109	2893	0.16

I have calculated the top 10 nodes for various centrality measures and I have also plotted the distributions of the coefficients.







Looking at the closeness graph, I think the trend may be related to the peak of the out-degree, in fact, in the network there are quite a high number of nodes with high out-degree, and these nodes can reach the other nodes more easily than the others. Another possibility can be how the numbers are how the numbers are considered equal, I mean that decimal numbers have many digits after the decimal point (about 15), maybe python cuts some of these digits and this makes the results agglomerate.