Exercise 1

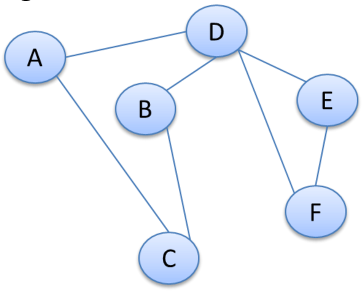
Exercises Peer-to-Peer-Systems and Security (SS2012)

Dr. Heiko Niedermayer
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München

Thursday 3.5. 2012
Hand-in: Thursday 10.5. in lecture or by mail to niedermayer – at – net.in.tum.de
Exercise: Monday 14.5. in lecture

Rules: There will be four exercise sheets. You have to hand-in 70 % of the assignments, attend at least 3 exercise courses and present a solution in the exercise course to get the 0.3 bonus.

Assignment 1 Clustering-Coefficient C and characteristic path length L
This assignment is about the clustering coefficient and the characteristic path length. Here is the graph.



Determine C, L as well as the diameter of the graph.

Solution:

A with neighbors C,D → C_A = 0 / 1 = 0 (both neighbors not connected)
B with neighbors,C,D → C_B = 0 / 1 = 0 (both neighbors not connected)
C with neighbors A,B → C_C = 0 / 1 = 0 (both neighbors not connected)
D with neighbors A,B,E,F → C_D = 1 / 6 = 0,167 (1 of 6 possible connections)
E with neighbors D, F → C_E = 1 / 1 = 1 (both neighbors connected)
F with neighbors D,E → C_E = 1 / 1 = 1 (both neighbors connected)

→ C = (0 + 0 + 0 + 1/6 + 1 + 1) / 6 = (2 + 1/6) / 6 = (13 / 6) / 6 = 13 / 36 = 0,361

A-B 2 A-C 1 A-D 1 A-E 2 A-F 2 B-C 1 B-D 1 B-E 2 B-F 2
C-D 2 C-E 3 C-F 3 D-E 1 D-F 1 E-F 1

→ L = (7 * 1 + 6 * 2 + 2 *3) / 15 = 25 / 15 = 5/3 = 1.67

D = 3 (e.g. C to F)

Assignment 2 Clustering-Koeffizient C

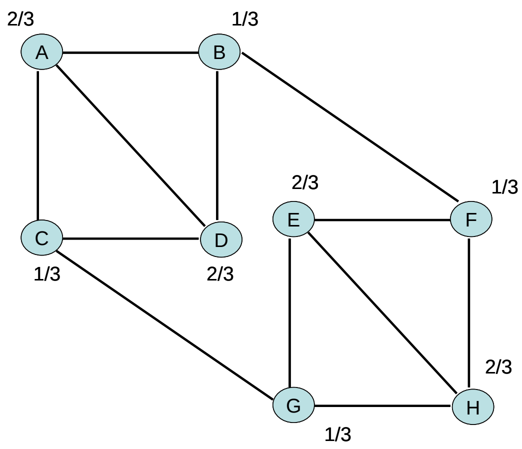
In this assignment you should create an example graph with certain properties.

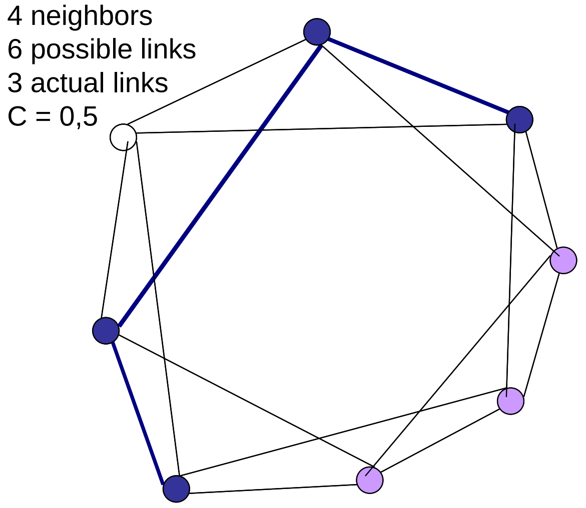
a) A connected graph with 8 nodes and C=0.5 (approx.). Prove your claim by calculating C.

b) A connected graph with 6 nodes and at least 7 links and C=0. Calculate C for the graph.

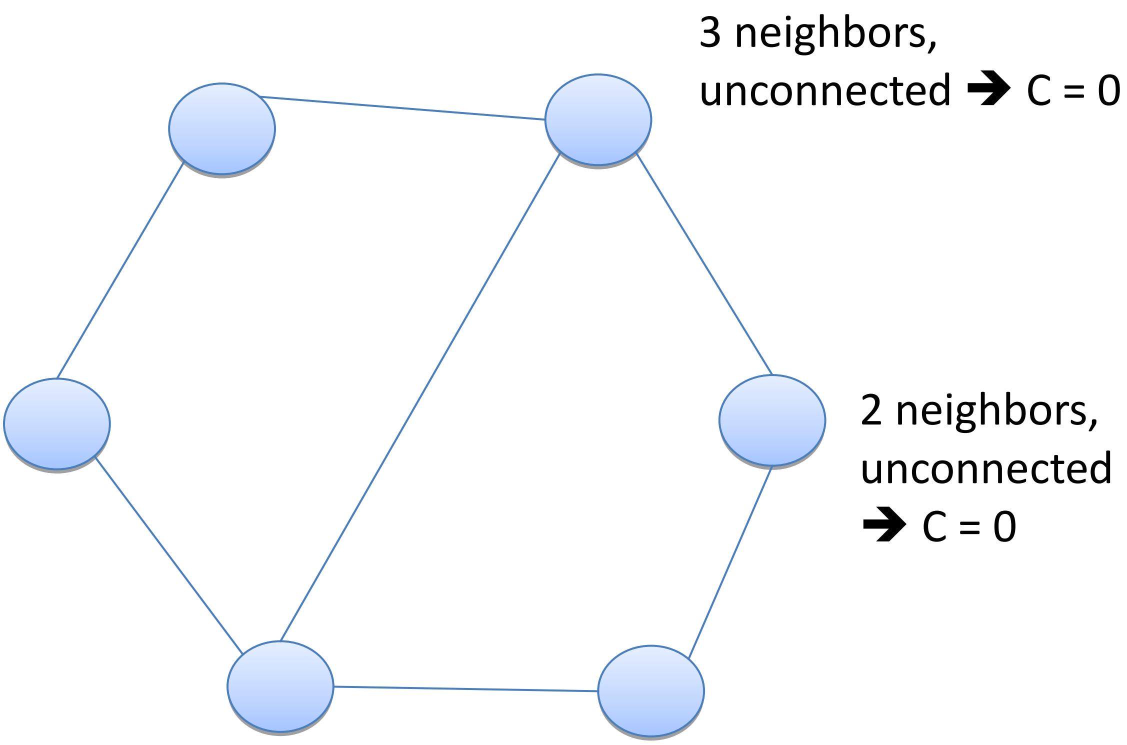
Solution:

a)





b)



Assignment 3 P2P Protocol

A protocol for an unstructured network. Each node joins via some node it knows. As a first step, it simply adds a link between itself and the node it used for the join. Then it operates as follows:

- If the node has less than 100 neighbors, every 100 s the node asks a neighbor for 10 other nodes. Each of the nodes in the reply will be contacted with probability p=25 %. This means that it will establish a connection, and add it to the list of known nodes. Otherwise, it ignores the node.
- Every 5 s it will contact one random neighbor to see if it still exists. If not, the connection to the node will be closed and the neighbor will be removed from the list of known nodes.

Questions:

a) What is the probability that none of the 10 nodes in a reply is contacted?
b) Give an example for a problem of this protocol. How could you fix it?
c) Assume the network is bootstrapped via one server that knows all nodes. If a new peer wants to join, it asks for a node that is in the network. The server returns a random peer. What kind of a network graph would you expect (Random, Small World, Power Law)? Argue why your answer is true (hint: state your assumptions and conclusions) and why the others are not.

Solution:

a) The node has to decide 10 times not to connect its possible neighbors, each time with $p_{no\_connection}$ = 0,75.

0.75^(10) = 0.0593 = 5.63 %

b)

- If the node has no additional contacts yet and his only known node leaves the net (first 100s). → no connectivity

- If a node only gets to know nodes that will leave soon and the average amount of nodes remains small→ there is always the risk to loose connectivity with the net.

- Slow assembly of neighbourhood. One could say, the cache is being filled more slowly than necessary.

The basic idea to fix the problem is to add all nodes first and use the probability only later-on. One should also ask for new neighbors directly in the beginning and not wait 100s before doing the first request.

Non-standard answers / understandings:

- Some people may consider the 100 nodes limit also for the incoming links. In that case one may run into the situation that new nodes cannot add any node to their links. Imagine 100 nodes in the network. All of them connected. Then node 101 will only find nodes with 100 links and no one will accept his connection request.

c)

The outcome is a bit parameter-dependent and scenario-dependent.

Random Graph: unlikely as node discovery is biased through existing neighbors… could be approximated if p is reduced extremely or only limited number of neighbors from direct neighbors are taken (measures to make it more likely to find and add distant nodes also).

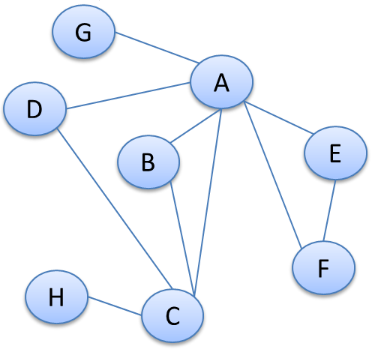
Small-World-Graph: as you add neighbors of neighbors, it is quite likely that you produce an increased clustering coefficient compared to a pure random graph. So, small-world graph is quite likely.

Power Law:

- Even though the outlinks are limited by 109 (99 nodes + 10 new nodes), the number of inlinks can vary a lot, in particular if the lifetime of nodes is varying a lot
- Only likely if parameters like node lifetime show Power Law / high variation properties.

Assignment 4 Power Law

Assume now that a new node N is joining the graph. It will set its first link according the Barabasi-Albert model (linear preferential attachment).



a) With what probability will it select node A?
b) Give an example of a reason that explains why a new node that does not know the network may have a higher chance of contacting node A than e.g. node G.
c) What will happen if an attacker shuts down node A?
d) What could you do to avoid Power Law graphs and hubs?

Solution:

a)

P_a = 6 / 20 = 0.3

b)

If the new node has no information about the network, then it cannot be explained. For the bootstrapping, however, some information has to leak. One example are server lists that can be downloaded from the website or are contained in the application. Nodes from such a list are much more likely to be contacted.

c)

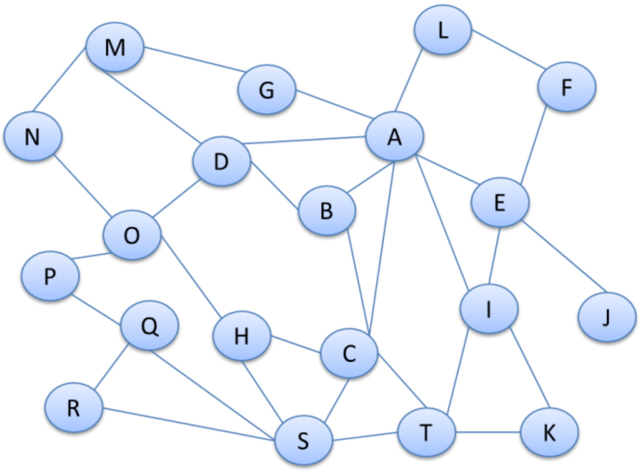
The network will be split into 3 partitions. So, the connectivity of the network will be lost.

d)

Prefer nodes with smaller degree when adding new links. A node that is asked for neighbors could e.g. select most of the nodes for the answer from nodes with low degree. Similarly, the node who gets these proposals could bias its selection by preferring nodes with lower degree. However, one should also add nodes with middle or high degree to avoid unstable nodes with low connectivity.

If nodes limit the number of inlinks, this may also help to avoid hubs. However, one has to ensure that nodes can create links and are not blocked.

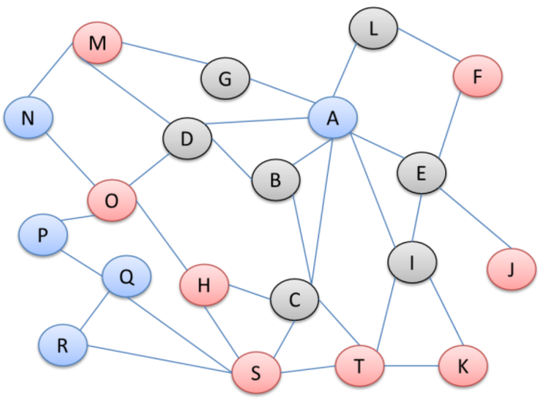
Assignment 5 CoolSpotsMunich I
Assume that the CoolSpotsMunich network is an unstructured network just as in the first slides of chapter 1.2 where the peers form an unstructured network and each peers stores its own data (items = "spots" / interesting points in Munich area with description, rating, GPS location, etc.) locally.
You will now optimize the network using replication of items. Every hour a node will look up 2 random peers using a random walk. At each hop a random next hop is selected (all previous hops excluded). Each walk is 3 hops long. You will do this 2 times to get the 2 peers.
The node will then store its spots (items) on these 2 peers. Nodes will delete the spots / items of other nodes after 2 hours.



Assume the lookup is limited to 2 hops (neighbour + neighbour of neighbour).

a) Node A will now ask for all spots in Garching. Can A be sure to get back all items on Garching that exist in the network? Justify your answer.
b) Assume that the random walks of node O ended on nodes A and T. Are its items now available to all nodes? How many nodes would reach its items without this replication?

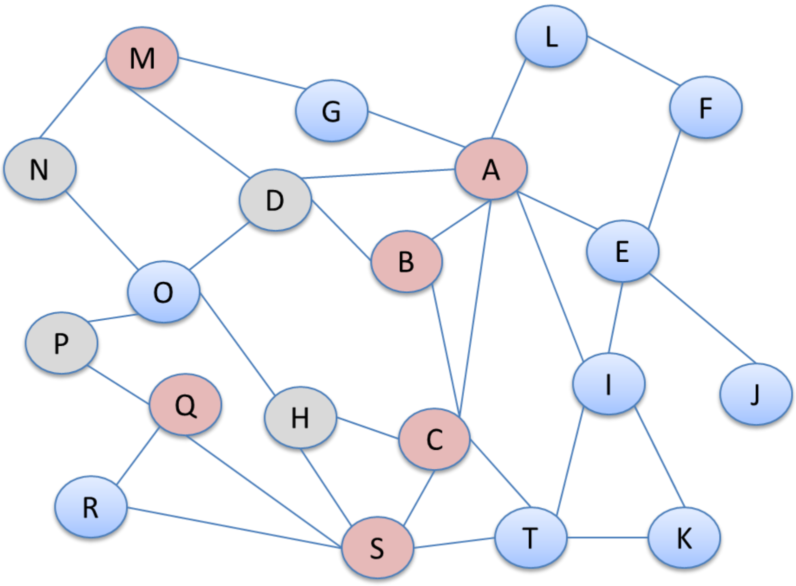
Solution:
a)
Take note: lookup is **2** hops, replication is **3** hops!



The 2 neighborhood of node A includes all nodes in the graph but N, P , Q, R. Dependent on how the replication is implemented exactly, it may be that one of them only replicates on them, e.g. both random walks go same way, a circle happens and we have to brake the rule not to take previous nodes.

b)

Given the 2 neighborhood of the nodes (e.g. node A from a)), all nodes will see the information.



The nodes R, T, K, I, J, E, F, L, G were not able to get the information without replication. 10 nodes besides A were able to rearch it before and 9 were not.