Peer to Peer Systems and Blockchains
Academic Year 2018/2019

Mid Term
Analysing the Kademlia DHT
Deadline 01-05-2019

The mid term consists in a study of the Kademlia Distributed Hash Table [1]. The first part of the assignment requires to write a simulation of the construction of the Kademlia routing tables. The simulation takes as input the number m of bits of the identifiers of the Kademlia network and the number n of nodes that will join the network. The routing tables of all the nodes are managed by a centralized coordinator which executes the following steps:

1. *initialization phase*
   it initializes a data structure of n elements which will contain the descriptors (including the routing tables) of the n nodes of the Kademlia network, as they will join the network. A first node, whose identifier is selected at random, is inserted in the data structure, with an empty routing table. This phase is executed only once.

2. *routing table construction*
   it picks up the identifier of a node p chosen at random, in the range of allowed identifiers, and simulates the join of p to the Kademlia network. This is repeated $n-1$ times, for all nodes joining the network, apart the first one.

   (a) it generates the identifier of a node, b chosen at random among the identifiers of the nodes already belonging to the network. b represents the *bootstrap node* of p.

   (b) the coordinator generates a sequence of identifiers ID, uniformly distributed, at random, in the identifiers range paired with the different buckets of the routing tables of p. For each ID, a $FIND\_NODE(ID)$ is sent to b, then the function is recursively invoked by exploiting the information returned by the previous call. This way, the routing table of p is filled with the information returned by the recursive invocations of the function $FIND\_NODE(ID)$.

   (c) The $FIND\_NODE$ message also keeps track of the nodes traversed. This way a node that receives a $FIND\_NODE$ message can update its routing table based on these identifiers

3. *analysis of the topology*
   After the joining of all the nodes is completed, the information contained in the routing tables is also stored on a file in order to be used by an external analysis tool. The information stored in the file is exploited to perform a set of analyses of the Kademlia topology. A *directed* graph is built from the routing tables in this way: if node A contains B in its routing table, an edge from node A to node B will be present in the graph. Then, it is requested

to perform a study of this graph and show, at least, the average degree of the network, its diameter and the clustering coefficient. Report the results of different simulation performed by varying m (number of bits), n (number of nodes) and k (size of the routing tables).

To this aim, it is recommended to use a graph analysis tool, such as: *Cytoscape* tool (http://www.cytoscape.org/), *Gephi* (https://gephi.org/), *Networkx* (Python only) (https://networkx.github.io/) or *Webgraph* (http://webgraph.di.unimi.it/).

The assignment requires the submission of:

- the code of the simulation. (It is recommended to define the simulation in JAVA, even if other languages are accepted). Code should be adequately commented.
- a brief report describing :
  - the main project choices
  - a set of plots showing a set of statistical measures of the Kademlia topology.

The assignment must be done individually and its deadline is 1 May 2019. If the evaluation of both the mid and of the final term will be positive, the student will be relieved from the oral exam. Submit the assignment through Moodle. Its evaluation will be notified through the Moodle as well.

The assignment is not mandatory, if it is not presented, the student will be required to pass the oral exam.

References

[1] *Kademlia: A peer-to-peer information system based on the xor metric*, Maymounkov, Petar and Mazieres, David, International Workshop on Peer-to-Peer Systems, 53-65 (2002),Springer.