Peer To Peer Systems and Blockchains

Analysing the Chord DHT

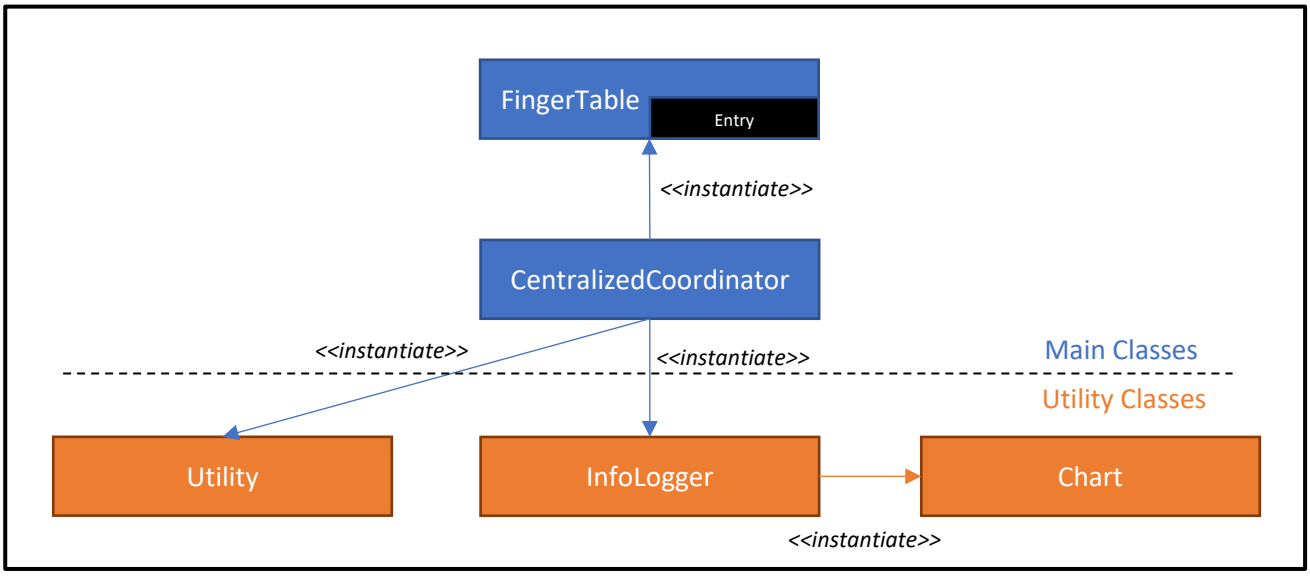
Mid Term Report

Academic Year 2017/2018

Alessandro Berti
505328

Contents

ARCHITECTURE



DATA STRUCTURES

The two main data structures that play a central role in the implementation of the simulation are:

- **ArrayList<BigInteger> nodes**: it contains the set of peers.
- **TreeMap<BigInteger, FingerTable> fingerTable**: which associate each peer to its finger table.

MAIN CLASSES

- ***CentralizedCoordinator***

  This class deals with the main function of the simulation:

  - It generates a set of peers so that they are mapped in the Chord ring
  - It generates a set of keys
  - It sets up the Finger Table for each generated peer
  - It executes the *LookUp* for each key previously created
  - It uses the **InfoLogger** class to generate charts based on the information collected during the simulation.

- ***FingerTable***

  It implements the finger table through a **List<Entry>**, where **Entry** is an inner private class of FingerTable class that contains two field: *target* and *link*.

  This choice permits to encapsulate **Entry** class, so that each interaction with this class is only handled by the set of method offered in FingerTable class, gaining in abstraction.

UTILITY CLASSES

- ***Utility***

  It offers methods for random IP and port generation and methods needed to compute the SHA-I on both nodes and keys.

  The only thing to stress is about SHA generation: since SHA returns a 160 bit value, it has been necessary to apply a module function to this value, in order to guarantee that each ID, that is going to be generated, will fall within the Chord Ring space (=2^{idBits}, where *idBits* is the length, in bit, of each ID). By doing so, in some sense, the implementation is closer to reality but, on the other hand, the simulation *loses the uniform distribution property*, guaranteed by SHA-I function.

  This consequence of this choice can be seen in the graphs below.

- ***InfoLogger***

  This class is used to manage the collection of datas during the simulation. Once the simulation has terminated, InfoLogger produces five charts: three about *statistical informatio*n (*e.g. hop needed to resolve a query*) and two about *Chord topology*.

- ***Chart***

  A simple class needed to simplify charts creation.

ANALYSIS

More simulations were performed with different input values; below are presented the charts obtained by the most significative simulation.

The charts computed by the other simulation are included in the attachment.

The following results were obtained from a simulation with:

- ***m** = **idBits** = 15*
- ***n** = **peers/nodes** = 5000*
- ***Queries** = 50000*

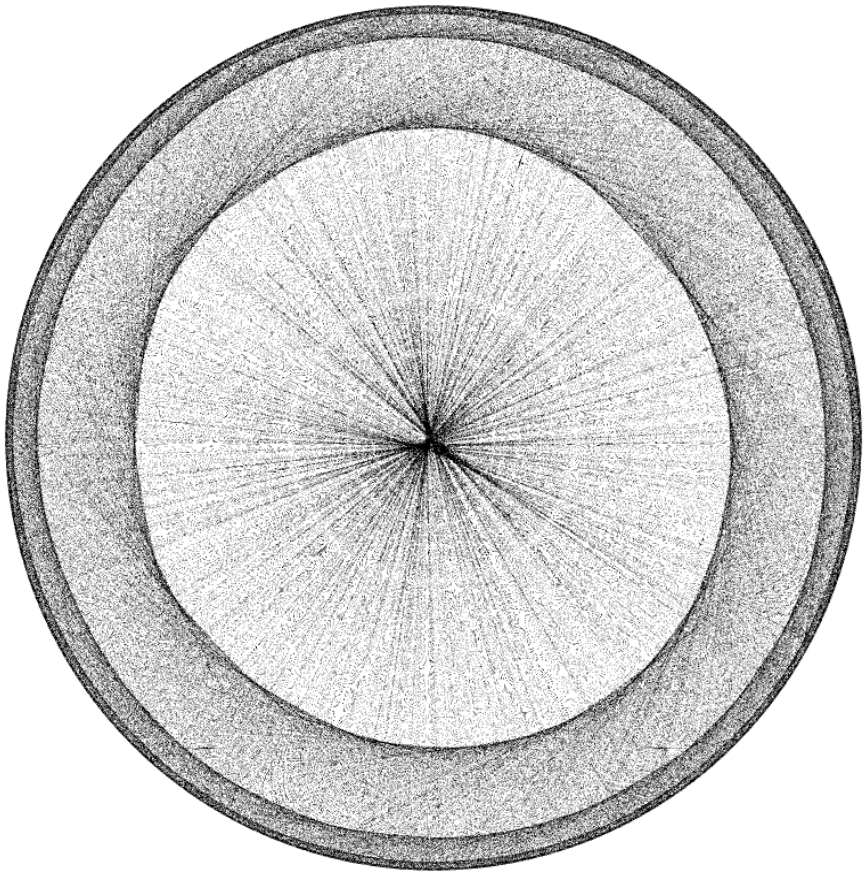


Figure 1. Chord Graph obtained from 5000 peers

STATISTICAL ANALYSIS

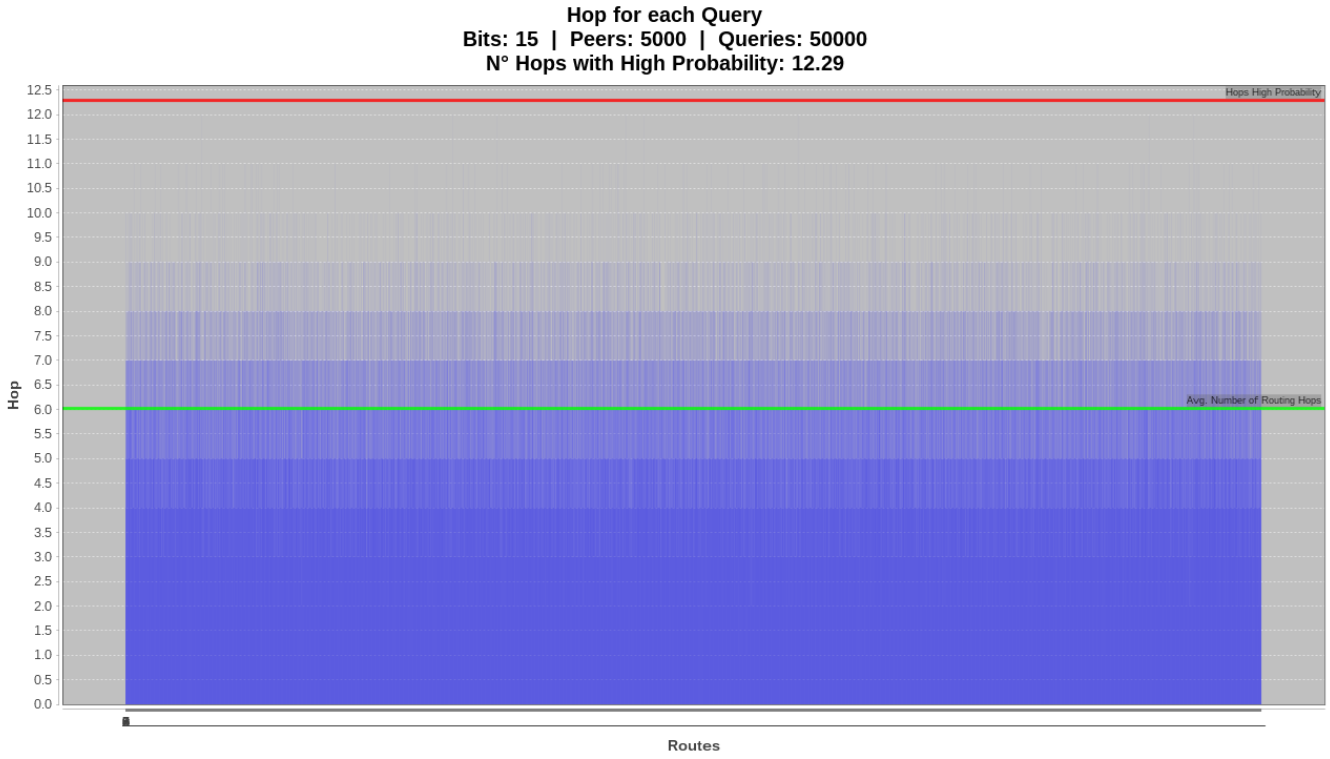


Chart 2. On the x axis, the number of hop. On the y axis, the times that a query has been resolved in x Hops.

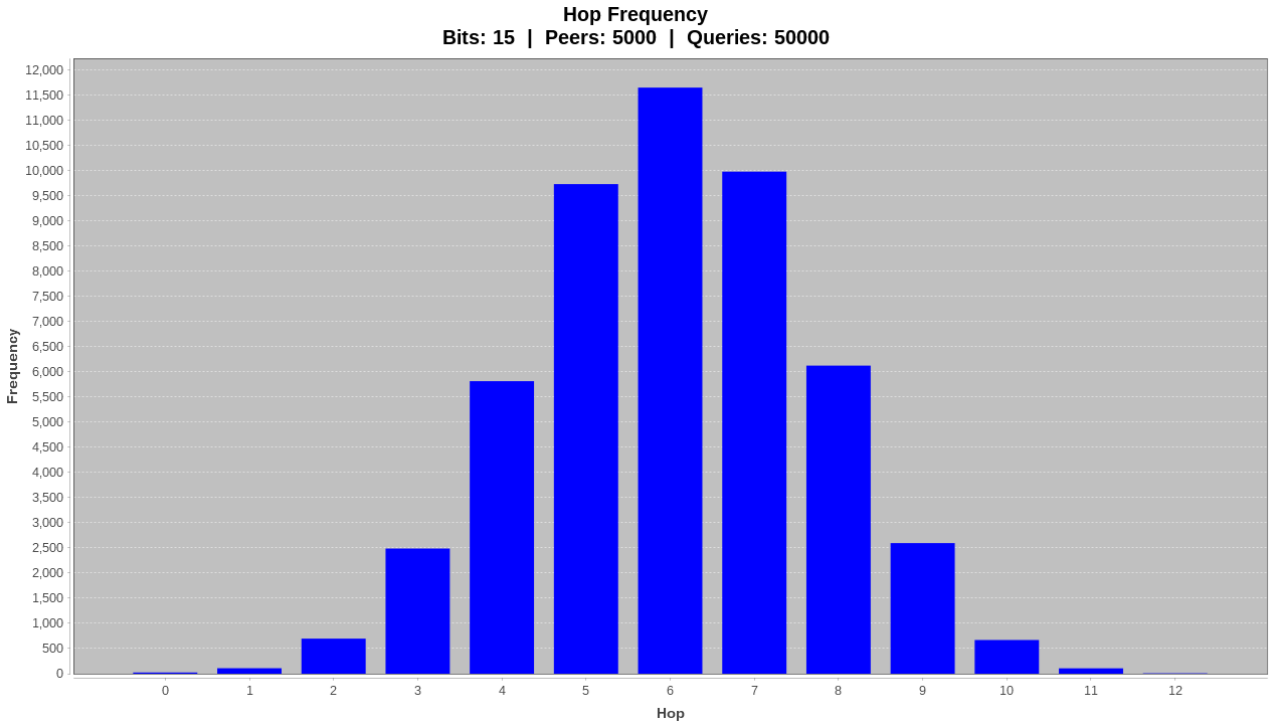


Chart 1. On the x axis, the query ID. On the y axis, how many hops to resolve the query.
Green Marker = Avg. Number of routing hops. Red Maker = N° Hops with High Probability

Chart 2 can be approximate by a *gaussian distribution* centred on value 6, that correspond to the mean of routing hops needed to resolve a query. We can observe that, Chart 2 is a better way to represent the same infos of Chart 1.

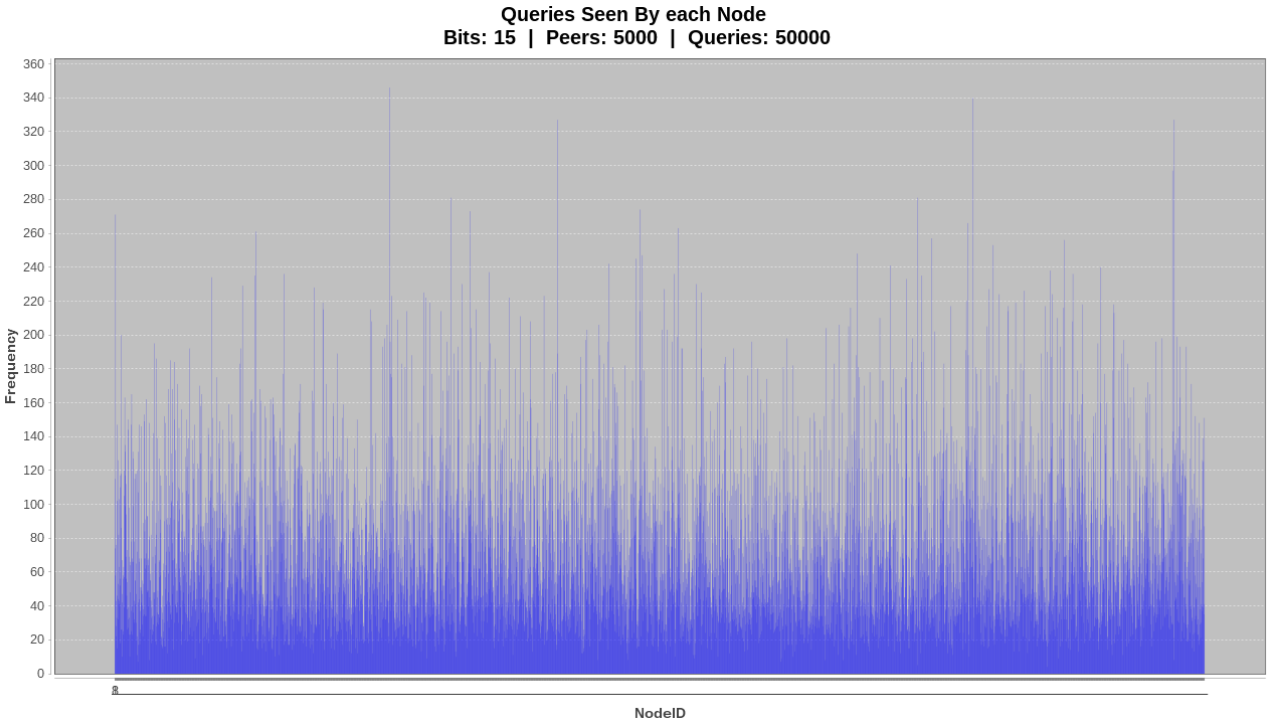


Chart 3. On the x axis, the nodes's id. On the y axis, how many times that node has forwarded a query.

Immediately, we can observe that the values in Chart 3 are *not uniform*. That implies that there are nodes that forward a higher number of messages than others. It reveals the *problem of load balancing* in Chord Routing; this intuition will be explicit observing the topological charts below.
Anyway, for the sake truth, in our case, this issue is aggravated by the lack of uniform distribution property, which has been lost due to the use of module function on a SHA-ed value.

TOPOLOGICAL ANALYSIS

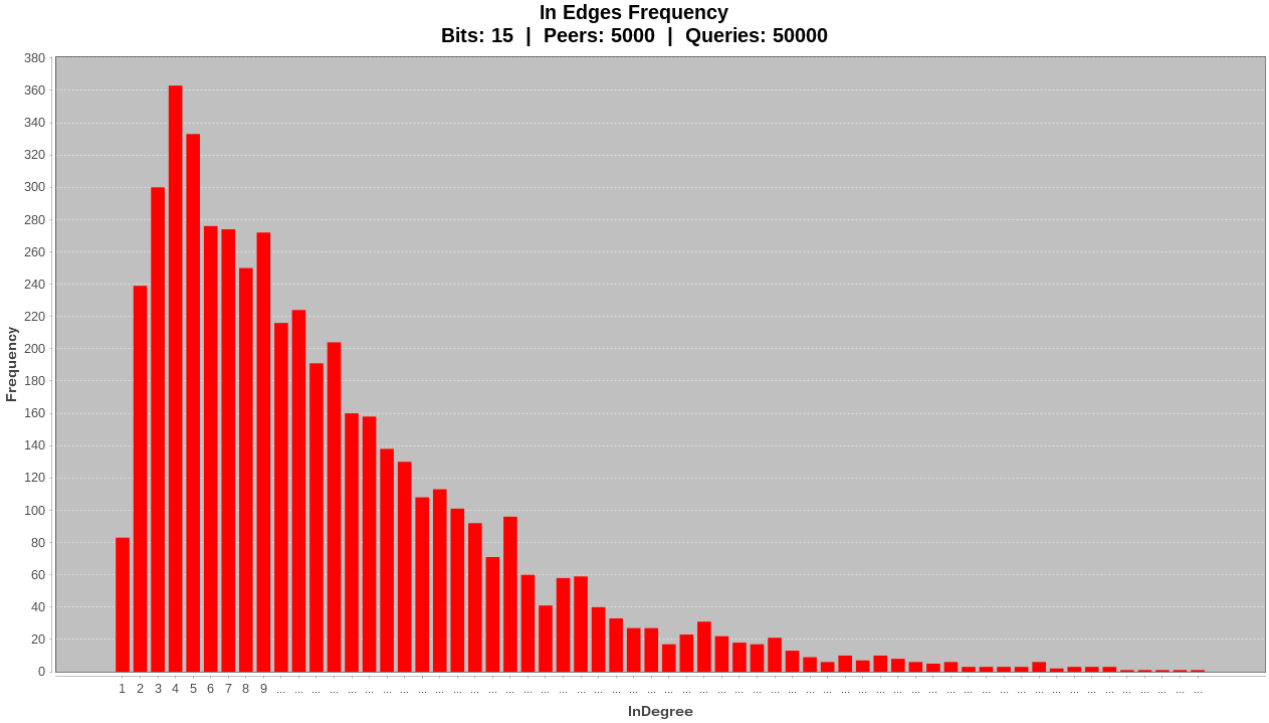


Chart 4. On the x axis the In-Degree value. On the y axis, how many nodes have exactly that number of incoming edges.

Chart 4 can be approximate by an exponential curve. This means that, since some nodes have a higher number of incoming arcs, then these nodes will be more likely to receive a query to forward.
So, if with the statistical analysis observed in Chart 3 had made us suspect about the presence of load balancing problem, the topological analysis of Chart 4 provides us a concrete proof about it.

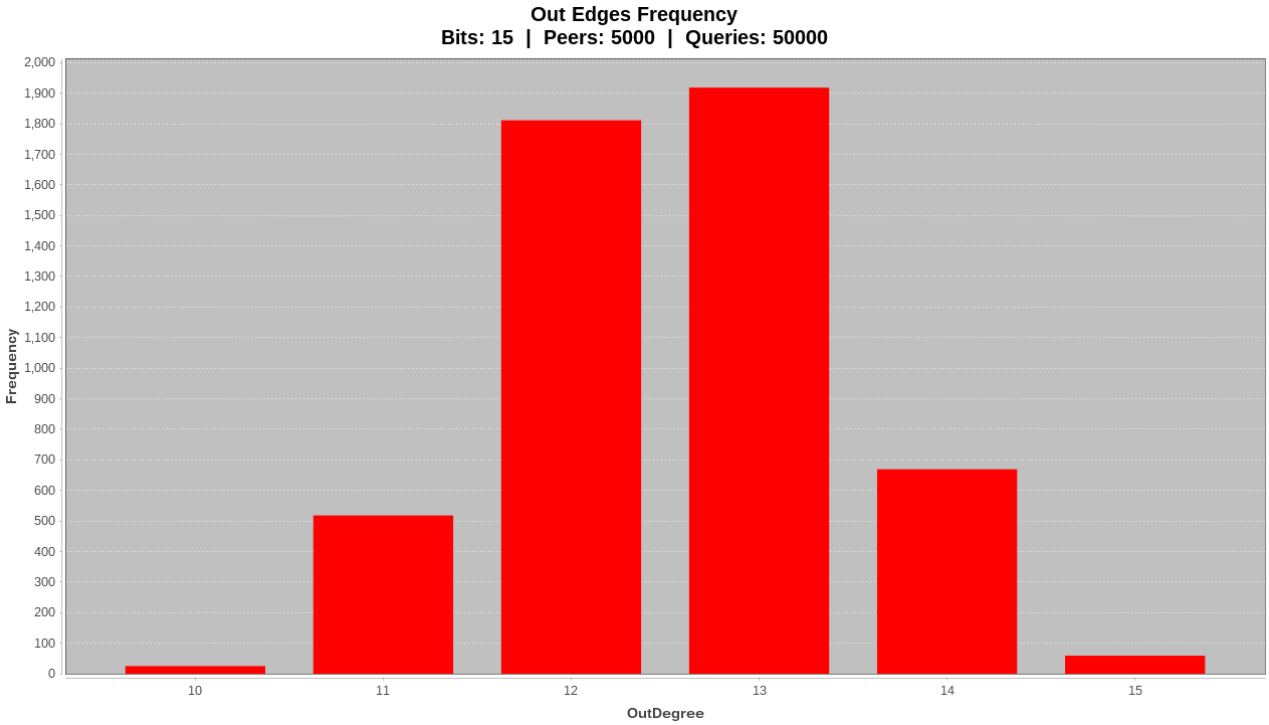


Chart 5. On the x axis the Out-Degree value. On the y axis, how many nodes have exactly that number of outgoing edges.

Chart 5 can be approximate by a *gaussian distribution,* with symmetrical values respect to 12.5. This means that each finger table has some repeated entries. In our case with m = 15:

- On average, around 2.5 entries are duplicated.
- The ***minimum*** number of non-duplicated entries in a finger table is 10.
- The ***maximum*** number of non-duplicated entries in a finger table is 15.
- The number of nodes that have 10 or 15 entries in their finger tables is quite low.

Observing other simulations, when $\lim_{peers \to 2^{idBits}} \left(\frac{peers}{2^{idBits}}\right) = E[All\ finger\ table\ entries\ are\ distinct]$.

That is: *to stretch the **number of peers** to the **cardinality of spaceIDs**, the **probability** that the entries of a finger table are distinct from each other, tends to 1.*

- ❖ intervals between peers become smaller and smaller
- ❖ the number of distinct entries increase
- ❖ the variance in Chart 5 decreases.

FINAL ANALYSIS

The Chord Overlay of this simulation has:

- Diameter = 10
- Avg. Path Length = 5.024
- Avg. Clustering Coefficient = 0.089
- Density 0.003

We observe that:

- the clustering coefficient is around 30 times the density, then the clustering coefficient can be considered high.
- Since the chord overlay has 5000 nodes and the diameter is 10, then the diameter can be considered small.

In conclusion, due to the low diameter and the high clustering coefficient, the Chord Overlay can be approximately described as a ***small world.***