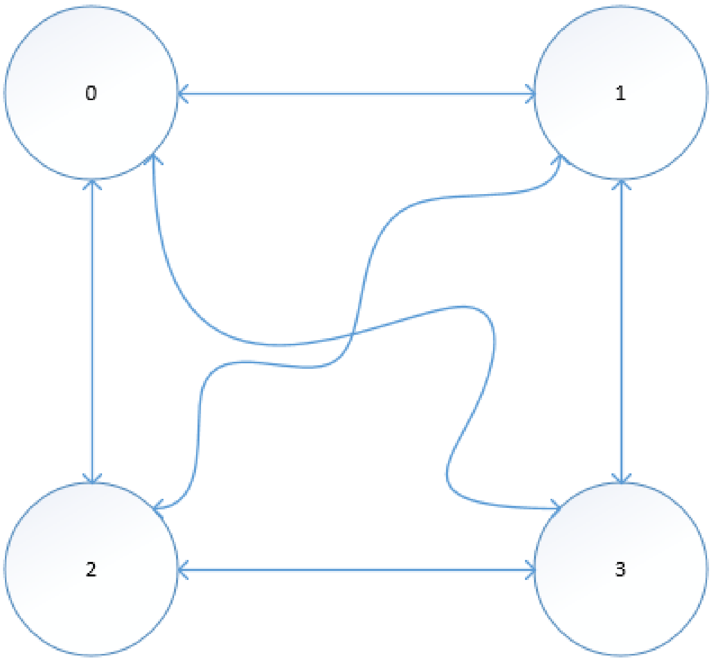
Distributed Artificial Intelligence and Intelligent Agents Homework 1

KTH Royal Institute of Technology
School of Information and Communication Technology
Student:Fanti Machmount Al Samisti (fmas@kth.se)
Student:Pradeep Perris (pradeep@kth.se)

February 11, 2016

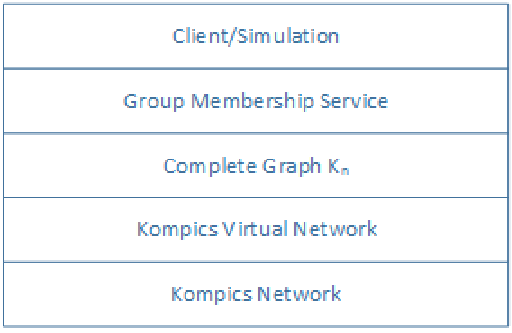
1 Introduction

The goal of this project is to implement a distributed key-value store with many freedoms given at hand like the structure, replication algorithm and factor and much more. We chose a full graph network e.g. if **n = 4** then K_4(in graph theory terms).



2 Enter the matrix

Before getting deeper into the architecture of our system, an overview picture is much more effective than any description:



Having the above in mind, we can move on to the insides of our system. A *Kompics* virtual network is comprised of *vnodes* that share the same physical traits, namely IP address and port number, and they are distinguished by an *id*.

2.1 Initialization

When then system is *bootstraping*, the single *NodeParent* component reads the following configuration file:

```
network {
    node {
        host = "127.0.0.1"
        port = 34567
    }
    grid {
        num = 3 # Size of the network, has to be at least 3
    }
}
```

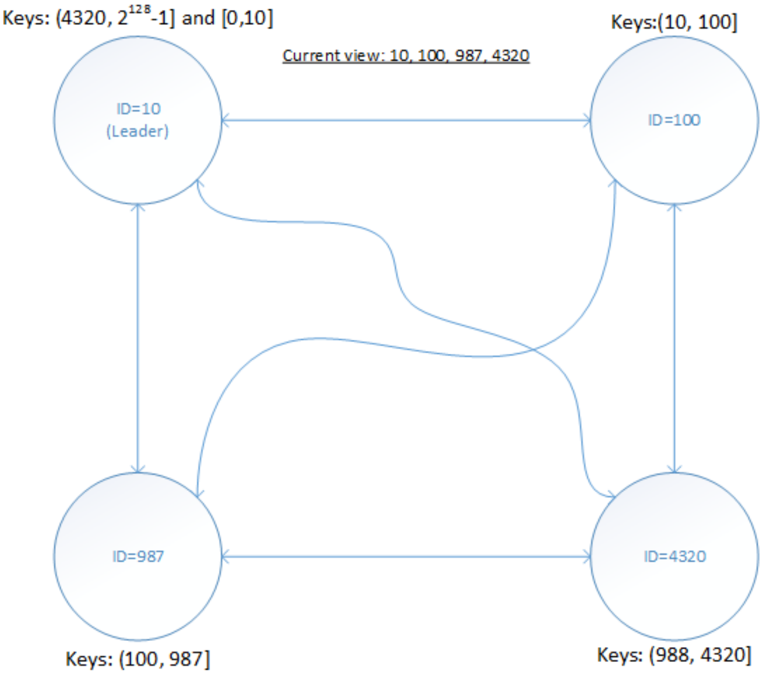
Having done that its time to start up the nodes and create our group with the *Node* components. The first node is always the initial group *leader*. After him, all the subsequent nodes are *slaves* and they have to send a *JOIN* message to the leader to get accounted for and retrieve an updated view of the group by receiving a *VIEW* message. Every node is tagged with a random number whose use will be explained in the next section.

A possible output of the initialization phase can be the following:

```
INFO  {Node} [0]: Got JOIN message from ID: [1]
INFO  {Node} [0]: Got JOIN message from ID: [2]
INFO  {Node} [1]: Got VIEW message from ID: [0](0 1)
INFO  {Node} [1]: Got VIEW message from ID: [0](0 1 2)
INFO  {Node} [2]: Got VIEW message from ID: [0](0 1 2)
```

2.2 Key assignment

We pretty much followed *Chord's* key assignment algorithm and applied it to our group. With a few nodes the key assignment will not be fair but as the size increases the distribution will be normalized.



2.3 Copy here, copy there, copies everywhere!

replication

2.4 Can we find the failures?

EPFD

2.5 Fix those crashes

maybe reconfiguration insight

3 References

- http://www.slideshare.net/WayneJonesJnr/chapter-16-distributed-system-structures-1314596

- http://blog.fourthbit.com/2015/04/12/building-a-distributed-fault-tolerant-key-value-store