

# SIMULATION OF PEER TO PEER NETWORK USING CHORD PROTOCOL IN NS2

Dr. Nader F Mir

Department of Electrical Engineering, San Jose State University, San Jose, California 95192

## Introduction

In the recent decade or so the peer to peer networks evolved from unstructured to more structured formats. The overlay networks used by the structured networks became more and more organized. Identification of the key is the prime element in any peer to peer protocol. The chord protocol does this efficiently and a quick and efficient mechanism in identifying the key of an element in a distributed environment.

### Motivation/Need and Application

The present day world is moving towards more of decentralization. The finest example to this is the bitcoin. It led the world with its peer to peer approach widely. For any of such approach to succeed, the efficient key location is of high importance and thus chord protocol find its importance in the present day world.

### Background

The chord protocol was evolved from a concept called Distributed Hash table. The circular architecture was an addition to Gnutella protocol in the peer to peer architecture

## Chord Protocol and its specifications

**Cooperative mirroring:** The Mirroring system would provide a balance load among all servers, cache and replicate them easily.

**Time-Shared Storage:** This helps for nodes with alternating connectivity.

**Large-Scale Combinatorial Search:** It is used in keys which are problematic. Chord helps in mapping these keys to the problematic machines.

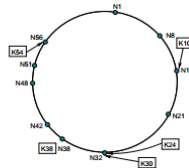
### Chord Protocol Specification:

- The chord protocol is one-dimensional, circular ID space.
- The number of peers at any time is  $N$ .
- The Hops to locate the data is regarded as  $\log N$ .
- The routing state at any given time is  $\log N$ .
- The case when peer joins and leaves is given by  $(\log N)^2$ .

## Key References

1. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications by Ion Stoica , Robert Morris , David Liben-nowell , David R. Karger , M. Frans Kaashoek , Frank Dabek , Hari Balakrishnan.
2. A scalable content addressable network - Ratnasamy, Francis
3. The Hybrid Chord Protocol: A Peer-to-peer Lookup Service for Context-Aware Mobile Applications by Stefan Zöls, Rüdiger Schollmeier, Wolfgang Kellerer, Anthony Tarlano
4. ChordNet: Protocol Specification and Analysis by Dennis Schwerdel, Matthias Priebe, Paul Muller and Peter Merz
5. Implementing a Distributed Peer to Peer File Sharing System using CHEWBACCA – CHord, Enhanced With Basic Algorithm Corrections and Concurrent Activation - Matthew Baker, Russ Fink, David Trimm, Adam Whisman

## Architecture of chord

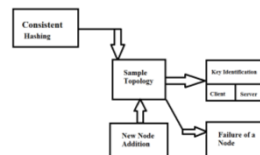


The chord protocol follows a circular architecture. The architecture is based on the Distributed Hash Table (DHT) architecture. Each node has knowledge of its successor and predecessor nodes. Thus making it simpler.

## Operations in Chord

- Start
- Join
- Delete
- Update
- Insert
- Get

### Block Diagram of the project



The block diagram explains the objective of the project. A sample topology is created in ns2 and each of node has its own elements. Each of the elements are of the node are hashed using SHA-1. The elements hashed are regarded as the key value pairs. The chord finger table is built for each node based on the topology size.

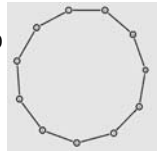
### Steps in the key identification

- A node is initiated and that node is regarded as the root node
- The successors and predecessors are negotiated.
- The finger tables are added generated.
- When a particular key is searched, the search pattern goes based on the finger table. For example node 0 it sends search query to the following nodes  
Node[0] => Node[1], Node[2] and Node[4]
- If success it returns the node location and the item are listed.

## Results

### Sample topology under experiment

The topology consists of 10 nodes in a circular architecture



### Key Value pairs

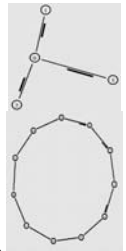
Each node has its own hash table. Where the filename is hashed and the hash value is store.

### Finger Table

The finger table is built as the routing path of each node. When a query arrives, the node searches the query by sending request to each node in its finger table.

Shown here is the query sent from node 0 to node 1 node 2 and node 4.

When a new node is added the nodes are stabilized, by sending updates.



### Observations

- The distant the key from the source node the longer time it takes to query.
- Node addition and deletion takes not much time stabilize the architecture.
- Not a standalone protocol must be combined with other protocols.

## Conclusions

The main proposed application of the chord protocol is to use chord based DNS. The idea behind this is to have node queries for domain names rather than have authoritative name servers for resolving hostnames. Some of the future enhancements of the chord protocol includes efficient node joining and leaving. Addition of security features to the chord. The inherent drawback of chord is it is not a standalone protocol. This has to be combined with other protocols to exploit its maximum strength.

## Acknowledgments

It is been a great privilege and a learning experience working on this project under my advisor Dr. Nader F Mir and I would like to thank him for his all-time support and cooperative suggestions.

## For further information

Please contact [srinidhi.hari@sjsu.edu](mailto:srinidhi.hari@sjsu.edu). The code and other details will be provided based on request.