
Programming Project 3

Gnutella-style peer-to-peer (P2P) file sharing system

Geoffrey Rathinpandi

R11488765

Contents

1	GithubLink	1
2	Overview	1
3	Architecture	1
3.1	Star Topology	1
3.2	Mesh Topology	3
4	Implementation	5
5	Future Enhancement	5

1 GithubLink

The github link : <https://github.com/GeOf3/Gnutella-peer-peer>

2 Overview

This project aims to provide a Gnutella -style peer-to-peer file sharing system. In which each peer will act as both client as well as a server. As a server, it accepts queries from other peers, checks for matches against its local data set and responds with results. As a client, it provides interfaces through which users can issue queries and view search results. In addition to this project since there is no central indexing server search is done in a distributed manner. Each peer maintains a list of neighboring peers whenever a query request comes in the peer will broadcast the query to all its neighbors in addition to searching its local storage and responds.

3 Architecture

There are two type of design implementation used in Gnutella

- Star Topology
- Mesh Topology

3.1 Star Topology

In star topology peer one is placed in the middle while the others are connected statically to peer one

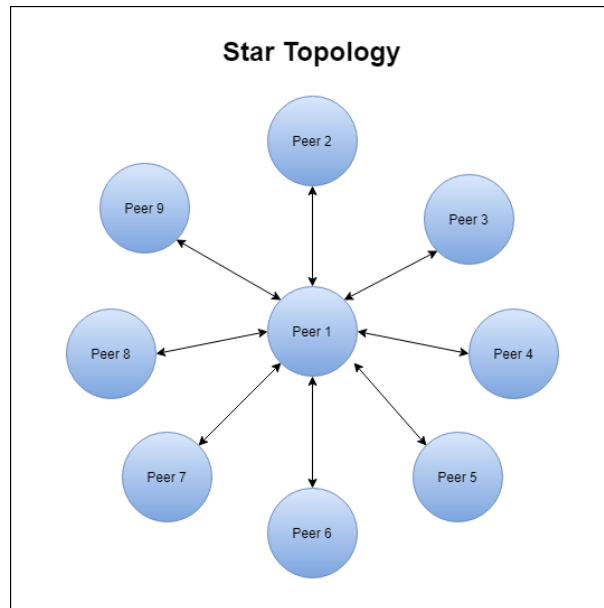


Figure 1: Star Topology

Config File

Below is the config file of Star Topology

```

p1 localhost 60000 p2-p3-p4-p5-p6-p7-p8-p9-p10
p2 localhost 60001 p1
p3 localhost 60002 p1
p4 localhost 60003 p1
p5 localhost 60004 p1
p6 localhost 60005 p1
p7 localhost 60006 p1
p8 localhost 60007 p1
p9 localhost 60008 p1
p10 localhost 60009 p1
+ 5 p5 localhost 60004 p1

```

Figure 2: Mesh Topology

3.2 Mesh Topology

In mesh topology peers are placed in a 2D-mesh and each peer is interconnected with one another.

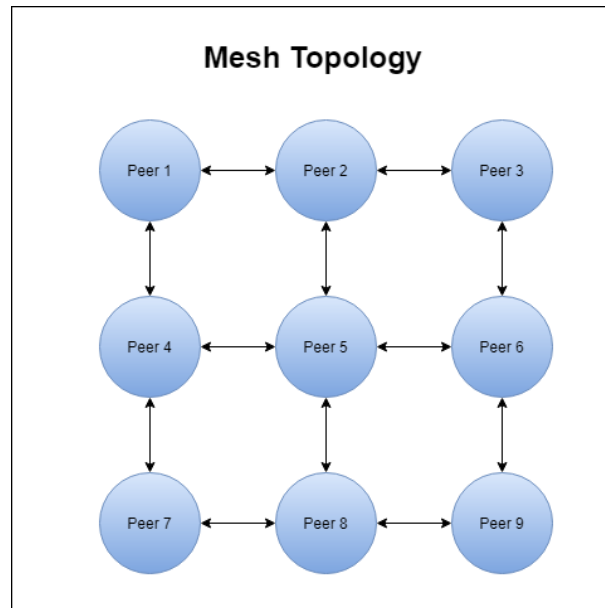


Figure 3: Mesh Topology

Config File

Below is the config file of Mesh Topology

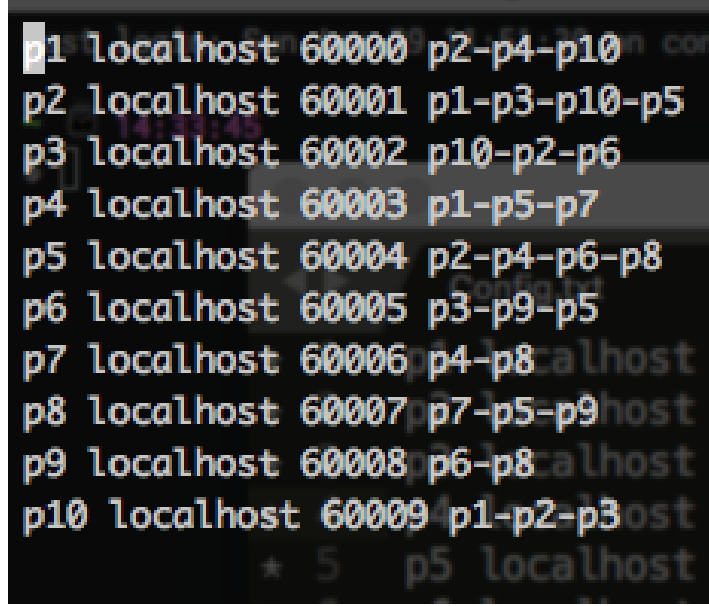


Figure 4: Mesh Topology

4 Implementation

When a user query a search request for a file, he will send a broadcast message to his neighbours who are going to do the same if they do not own the requested file. If a file found, the file owner will send back a hit query the requester through the addresses that have been stored in the header message of all passby nodes. If a node received the same message one more time, it will drop it immediately. This action is done by doing the following steps:

- 1. When a new message created, attache the PK (peer ID+time stamp) along with message, and store it in the PK field.
- 2. For any broadcasted message, record the PK in a list.
- 3. For every received message, compare the PK with the stored PKs. If found, drop the message.

Else, broadcast the message.

5 Future Enhancement

- Enhancing the performance by apply load balancing approach on the server side.

- Dynamic ports allocation needed in the system to avoid overlapping reserved ports.
- A simple GUI will make it user friendly.