LAB 4 [ECE 658]

Implementation of Caching in Unstructured Peer to Peer Networks

Detailed design Report

Eshwar Reddy Pasula (830531654), Gayatri Pendharkar (830670002)

1. INTRODUCTION:

The main purpose of this assignment is to devise a solution for searching contents in a network. This network is a basic distributed application layer network. It is a peer to peer network with a distributed network architecture. The nodes in the network serve the purpose of both client (firing queries for search) as well as server (handling the received requests). This is a much beneficial approach than centralized-server architecture where the server has all the resources and the individual nodes request for resources. However, in a peer-to-peer network, queries like searching for files are fired amongst multiple interconnected peers. Each of the nodes has its own share of resources. Whenever a request is received, the node makes a portion of their resources directly available to other network participants, without the need for centralized coordination by servers.

2. DESIGN OF THE NETWORK TOPOLOGY:

According to the problem statement, the implementation involves two main phases:

- **Implementation of the unstructured p2p network topology**
- **Design and develop a solution to search contents**

- A BootStrap Server(BS) is maintained which maintains the connections and each node is registered with the BS before it enters the network. Once the node is registered with the BS, the BS returns the information about the nodes (max 3) if any are previously registered under the same username. All the nodes communicate using UDP.
- Each node maintains a Routing table which will store the IP and port of the nodes its connected to, a hostname-resources.txt file with names of all the resources it contains and a log files which will keep a track of all the requests sent and responses received.
- The registered node then sends a join request to the nodes returned by the BS and stores the information returned by BS in the Routing table. Once the nodes receive join message, they store the information of the respective node in their Routing table.
- Once the request is sent by the current node to join or leave, the client starts the timer and waits for 2 seconds for an acknowledgement. This is implemented to avoid the client wait if packets get dropped due to the use of UDP.

- Accordingly, the network is formed with each node connected to at least 3 other nodes. If any node fails or wants to leave the network, it is unregistered from BS and is deleted from the Routing table of its neighbors. Leave message can be given anytime during the running time of the program.
- Once the network is formed, a search queries are fired from any/some/all nodes as per requirement. The search query can either be the entire message or a part of message (Partial search).
- When the connected nodes receive the search request, it will search amongst its resources for the requires file. If the file is present, the node will send the acknowledgement to the node requesting the file. If not, it will just forward the request in the network. Each time the request is forwarded in the network, the hop count is incremented.
- Node that initiated the request waits for a particular time period to receive responses from all of the nodes in network and times out and proceeds to next request. The number of hops and the time required to search the query will demonstrate the popularity of the query in the network. The topology will be simulated for 20,40,80 nodes in the network. Following diagram represents the network topology.

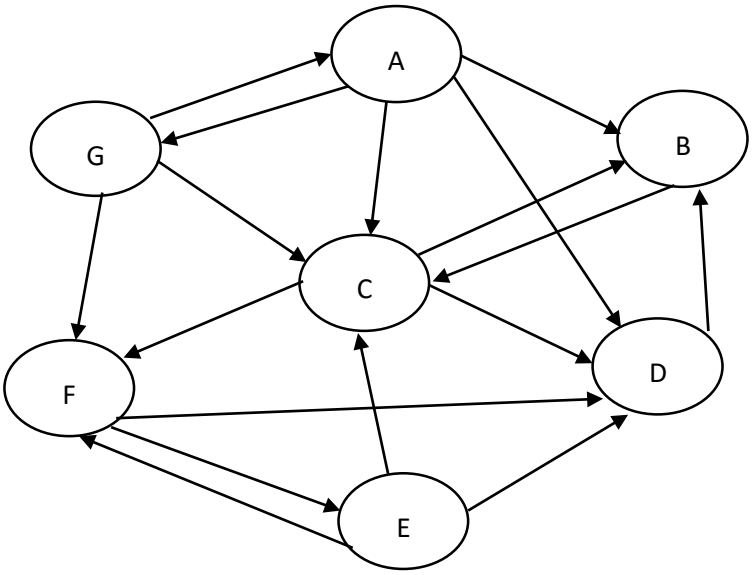

Fig. Unstructured p2p network representation for 7 nodes in network

- The figure above shows an unstructured p2p network with each node connected to at least 3 other nodes in the network.
- The most popular query will be the one which would be present at several nodes and could be acquired with less number of hops and the least popular one would be present with less nodes (at least one) and hence could take more number of hops to reach. This will be determined based on the zipf distribution.
- The search queries will be fired one after the other without waiting for any acknowledgements from queries.txt file.

3. CACHING SCHEME:

- An efficient caching scheme is to be implemented wherein each node will cache the location for the most popular query. At a given instance, any random Ns nodes will fire a query in the network.
- After the query is fired from these nodes, each node will parallel decide the most popular query amongst all the search requests it received. Depending on the responses it receives, it will store the responses for the top Nc searched queries in its cache table (Nc – size of caching table).
- Each time a new search query is received, the node will check for the most popular query and cache the data received from the responses respectively.
- By implementing this caching scheme, the nodes will have the entry for the queries most searched at that node. Thus, every time a new search request comes, it will already be cached hence reducing the latency and hop count required to search the resource.

4. CONCLUSION:

From the implemented p2p network, certain conclusions can be deduced. P2p networks behave quite differently from the centralized networks. As in a centralized server network, all the requests from clients are served by a centralized server. However, in current implementation, each node has a capability to act like both client and server. This is called de-centralized mode of communication.

Also, we have evaluated the performance of unstructured peer to peer network using different parameters like hop count and searching time for the query. Different functions have been implemented for different purposes like Server, Client, Searching, etc. which is a multi-layered stack implementation.