Design Document

CS-4262 Distributed systems

Distributed Content Searching

Group Members:

140033P Anees

140306G Kirisanth

140494D Ramiya

Suitable Topology

A peer to peer (p2p) file sharing system can be implemented using structured p2p or unstructured p2p overlays. We analyzed both overlays and chose Gnutella - unstructured p2p overlay as the suitable topology for this project.

- Connectivity

Unstructured overlay topology does not restrict the number of connections but it is generally more than two nodes. It is highly resilient to failure of nodes since it connects multiple nodes as neighbors. Even if we remove nodes randomly, system can work without any issues, thus it supports high failure rates. Early systems, such as Napster, use a central server to store indices of all peers in the network. This centralized design and practice arouse the concerns of performance bottleneck and single point of failure. Instead of maintaining a huge index in a centralized system, a decentralized system such as Gnutella distributes all searching and locating loads across all the participating peers.

- Availability

Unstructured topology provides high availability since it has more connections among the nodes. However this decentralized approach addresses the overloading and reliability issues to broadcast messages across a large population of peers. But considering the given requirements we assume the number of nodes is small (less than 100).

- Query success

The network uses flooding as the mechanism to send queries across the overlay with a limited scope. When a peer receives the flood query, it sends a list of all content matching the query to the originating peer. While flooding based techniques are effective for locating highly replicated items and are resilient to peers joining and leaving the system.

So, by considering above observation unstructured overlay (Gnutella) is suitable overlay topology to implement this file sharing system.

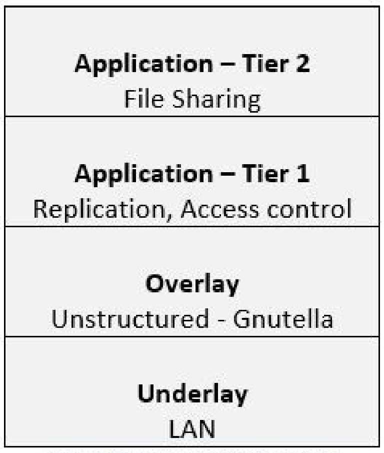


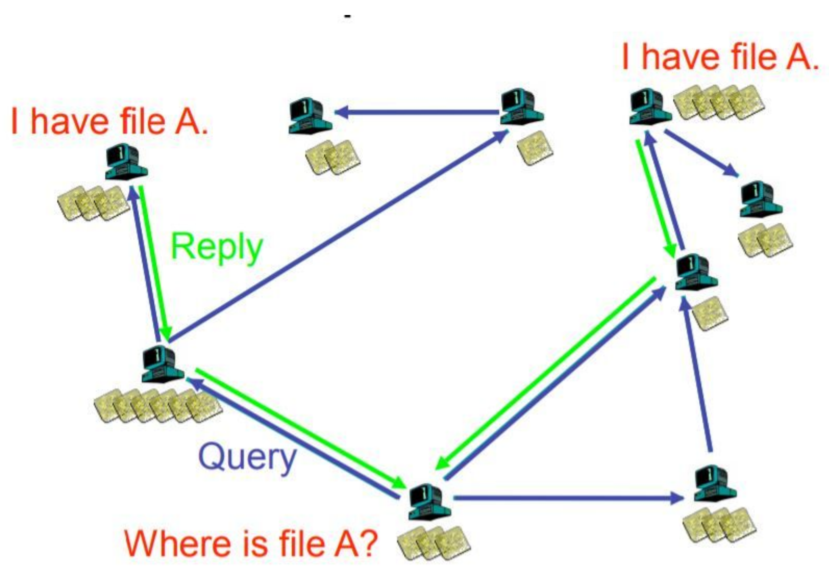
Figure 1 : Layered Architecture

Communication Among Nodes

There is no centralized servers exist in this unstructured overlay. Each Gnutella node only knows about the nodes with which it directly connects. That is each peer only have information about its neighbors. The other nodes are invisible, unless they announce themselves answering a message like ping or answering a query.

The access to the network can be made indicating IP address and TCP port of any node that already is connected. A new node that comes up has to be connected to 2 randomly selected nodes existing in the distributed system given by Bootstrap Server. A new node that initiates the connection announces its presence by sending a message by broadcast with neighbors. Removal or failure of individual peers can not impact on the system ability to transmit messages since all possible routes in the specified neighborhood are utilized simultaneously

The major existing mechanism for message broadcasting is flooding, in which a peer sends a message to its neighbors, which in turn forward the message to all their neighbors except the message sender. The act of query flooding can ensure guaranteed discovery but entails a large overhead traffic in the network. But TTL-based flooding can improve the performance by reducing the overhead of messages.



Performance Parameters & Measurements

- Hobs number
    - Counted by Time-to-Live (TTL) counter for each query

- Success rate
    - Success rate = # successful hits / # total searches
    - Calculate using test queries.

- Search responsiveness
    - Search responsiveness = success rate / hops number

- Latency
    - Latency = result delivery time - query trigger time
    - Use local timestamp at requesting node.

- Response time for graceful departures of nodes
    - Disconnect nodes randomly and measure time to respond.

- Query efficiency
    - Query efficiency = # query hits / msg per node

- Search efficiency
    - Search efficiency = query efficiency * search responsiveness